

Хмельницький національний університет  
Факультет програмування та  
комп'ютерних і телекомунікаційних систем  
Кафедра комп'ютерної інженерії та системного програмування

ДИПЛОМНА РОБОТА МАГІСТРА

Галузь знань \_\_\_\_\_ 12 – Інформаційні технології \_\_\_\_\_

Спеціальність \_\_\_\_\_ 123 – Комп'ютерна інженерія \_\_\_\_\_

на тему «Кіберфізична система виявлення просторових аномалій руху суден на основі методу кластеризації маршрутів»

ДРКІСПр. 015098.19.02.05 ПЗ

Виконав: студент 2 курсу, група КІ2м-19-1



Підпис

Костельна Т.В.

Ініціали, прізвище

Керівник доктор техн. наук, професор  
Науковий ступінь, вчене звання



Підпис

Боровик О.В.

Ініціали, прізвище

До захисту допускаю:

Зав. кафедри КІСП, д.т.н., проф.

Т.О. Говорущенко

25 05 2021 р.

Хмельницький, 2021

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ПРОГРАМУВАННЯ ТА КОМП'ЮТЕРНИХ І ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА СИСТЕМНОГО ПРОГРАМУВАННЯ

Освітній рівень МАГІСТР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЬО-НАУКОВА ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко

“ 03 ” 09 2020 р.

**ЗАВДАННЯ  
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)**

Костельній Тамарі Валентинівні

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Кіберфізична система виявлення просторових аномалій руху суден на основі методу кластеризації маршрутів

Керівник проекту (роботи) Боровик О.В., д.т.н., професор

Прізвище, ім'я, по батькові, науковий ступінь, місце зв'язку

Затверджена наказом ректора університету від 01.09.2020 р. № 119

2. Строк подання студентом проекту (роботи) на кафедру 05.05.2021 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Аналіз існуючих методів і засобів виявлення просторових аномалій руху суден

Метод кластеризації маршрутів руху суден

Алгоритми та технології реалізації удосконаленого методу кластеризації маршрутів руху суден

Кіберфізична система виявлення просторових аномалій руху суден на основі методу кластеризації їх маршрутів

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

\_\_\_\_\_


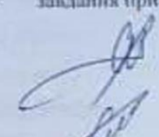

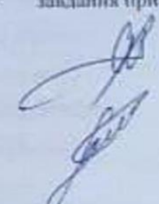
\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

6. Консультанти розділів дипломного проекту (роботи)

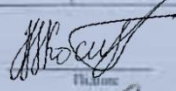
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КІСП		
Антиплагіат	Нічепорук А.О., старший викладач кафедри КІСП		

7. Дата видачі завдання « 03 » вересня 2020 р.

**КАЛЕНДАРНИЙ ПЛАН**

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики ДРМ з керівником	05.09.2020	виконано
2	Ознайомлення з предметною областю; формулювання мети та завдань дослідження; визначення об'єкта та предмета дослідження	05.10.2020	виконано
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	05.11.2020	виконано
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	05.12.2020	виконано
5	Робота над науковою статтею	12.01.2021	виконано
6	Робота над розділом 3 – розробка методів для вирішення поставленої задачі	20.01.2021	виконано
7	Робота над розділом 4 – проектування та розробка ПЗ для вирішення поставленої задачі, експериментальна частина	15.02.2021	виконано
8	Оформлення пояснювальної записки згідно вимог	23.03.2021	виконано
9	Попередній захист ДРМ	29.04.2021	виконано
10	Захист ДРМ на засіданні ЕК	До 04.05.2021	

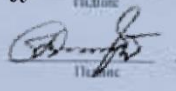
Студент



Костельна Т.В.

Ініціали, прізвище

Керівник проекту (роботи)



Боровик О.В.

Ініціали, прізвище

## РЕФЕРАТ

Тема дипломної роботи: кіберфізична система виявлення просторових аномалій руху суден на основі методу кластеризації маршрутів

Автор роботи: Костельна Тамара Валентинівна

Керівник роботи: Боровик Олег Васильович

Пояснювальна записка: 134 с., 21 рис., 1 табл., 3 дод., 70 джерел.

Ключові слова: метод, кластеризація, маршрут, система висвітлення надводної обстановки, просторові аномалії, еталонний.

Об'єктом дослідження є: процес обробки даних у системі висвітлення надводної обстановки.

Предметом дослідження є: науково-методичний апарат обробки даних у системі висвітлення надводної обстановки.

Метою дипломної роботи є розробка кіберфізичної системи виявлення просторових аномалій руху суден на основі покращеного методу кластеризації маршрутів.

Наукова новизна отриманих результатів полягає в тому, що удосконалений метод кластеризації маршрутів руху суден на основі побудови еталонного маршруту спрощує виявлення просторових аномалій руху суден, підвищує достовірність результатів моделювання та збільшує ефективність автоматизованої системи обробки даних СВНО.

На основі проведених досліджень розроблена кіберфізична система виявлення просторових аномалій руху суден на основі методу кластеризації.

Практична значимість отриманих результатів полягає у покращенні висвітлення надводної обстановки та контролю руху суден.

# ЗМІСТ

Скорочення та умовні позначки .....	7
Вступ.....	10
1 Аналіз існуючих методів і засобів виявлення просторових аномалій руху суден .....	10
1.1 Постановка задачі .....	10
1.2 Існуючі методи виявлення просторових аномалій руху суден.....	10
1.3 Існуючі засоби виявлення просторових аномалій руху суден.....	19
1.4 Висновки.....	26
2 Метод кластеризації маршрутів суден.....	27
2.1 Новітні підходи до оцінки відхилення маршрутів суден .....	27
2.2 Удосконалений метод кластеризації маршрутів руху суден .....	35
2.3 Висновки.....	45
3 Алгоритми реалізації удосконаленого методу кластеризації маршрутів суден.....	46
3.1 Алгоритми реалізації удосконаленого методу кластеризації маршрутів суден .....	46
3.2 Проектування програмного забезпечення реалізації удосконаленого методу кластеризації маршрутів руху суден.....	46
3.3 Висновки.....	55
4 Кіберфізична система виявлення просторових аномалій руху суден на основі методу кластеризації їх маршрутів.....	56
4.1 Опис середовища розробки кіберфізичної системи виявлення просторових аномалій руху суден.....	56
4.2 Програмна реалізація виявлення просторових аномалій руху суден .....	63
4.3 Висновки .....	67

Висновки .....	68
Перелік посилань.....	69
Додаток А Код (лістинг) програмного забезпечення .....	745
ДодатокБКопія тез.....	114
ДодатокВ Презентація по темі.....	127

## **СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ**

ПЗ - програмне забезпечення

ОСД-оперативно-службова діяльність

СВНО - система висвітлення надводної обстановки

КС – кіберфізична система

ІТС – інформаційно-телекомунікаційна система

## ВСТУП

Актуальність роботи. Важливим чинником стабільності держави є надійна охорона кордону. В сучасних умовах постійного зростання обсягів даних, які необхідно враховувати при прийнятті рішень, надзвичайно важливою є інформаційна складова сучасної моделі охорони кордону. Це стосується як наземної, так і повітряної і морської систем які забезпечують безпеку країни. В даній роботі буде розглядатися саме система контролю руху морських суден. Адже забезпечення цілісності морських кордонів України є одним із пріоритетів, особливо в теперішній час. Був проведений аналіз існуючих систем і методів оцінки і контролю надводної обстановки і завдяки цьому було визначено, що доступний на даний момент метод кластеризації маршрутів суден (який реалізований в системі «Гарт-12») потребує удосконалення. Робота присвячена аналізу існуючого методу кластеризації маршрутів суден з позиції обґрунтування можливості його застосування або визначення шляхів його удосконалення для використання в системі висвітлення надводної обстановки.

У результаті проведеного дослідження встановлено, що існуючий метод не може у явному вигляді бути застосовним для використання в системі висвітлення надводної обстановки як процедурний модуль автоматизованої системи обробки даних, який би забезпечував достатній рівень достовірності виявлення ознак порушення прикордонного законодавства.

Метою дипломної роботи є саме удосконалення і покращення даного методу. А також розробка системи де б був реалізований даний метод.

Поставлена мета досягається розв'язанням таких основних задач:

- здійснення аналізу існуючого методу кластеризації маршрутів суден в системі висвітлення надводної обстановки та виявити його «слабкі» місця;
- удосконалення методу кластеризації маршрутів суден, як процедурного модуля автоматизованої системи обробки даних у системі висвітлення надводної обстановки;

- розробка кіберфізичної системи для реалізації удосконаленого методу кластеризації маршрутів суден, як процедурного модуля автоматизованої системи обробки даних у системі висвітлення надводної обстановки.

Об'єктом дослідження є процес обробки даних у системі висвітлення надводної обстановки.

Предметом дослідження є науково-методичний апарат обробки даних у системі висвітлення надводної обстановки.

Наукова новизна отриманих результатів:

- удосконалено метод кластеризації маршрутів суден як процедурний модуль автоматизованої системи обробки даних у системі висвітлення надводної обстановки;
- набула подальшого розвитку інформаційна технологія кластеризації маршрутів суден у системі висвітлення надводної обстановки.

Практична цінність отриманих результатів. В результаті виконаного наукового дослідження розроблена кіберфізична система виявлення просторових аномалій руху судна на основі удосконаленого методу кластеризації даних про маршрути руху суден.

По темі наукової роботи опублікована стаття у Міжнародному науковому журналі «Комп'ютерні системи та інформаційні технології».

# 1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ І ЗАСОБІВ ВИЯВЛЕННЯ ПРОСТОРОВИХ АНОМАЛІЙ РУХУ СУДЕН

## 1.1 Постановка задачі

Надійний захист та охорона кордону є одним з головних факторів стабільного розвитку держави. Основними завданнями у галузі забезпечення прикордонної безпеки є гарантування недоторканності державного кордону та захист суверенних прав України в її виключній (морській) економічній зоні, а також покладання на Державну прикордонну службу України (ДПСУ), співробітники яких використовують систему пунктів технічного спостереження, дані з яких використовуються для спостереження та аналізу прикордонної ситуації.

У сучасних умовах, через постійне зростання обсягів даних, які необхідно враховувати при прийнятті рішень, найважливішою є інформаційна складова сучасної схеми охорони кордону, яка, в свою чергу, реалізується за допомогою інтегрованої інформаційно-телекомунікаційній системі (ІТС) «Гарт».

Особливості оперативно-службової діяльності (ОСД) органів та підрозділів Державної прикордонної служби України обумовлюють необхідність використання інформації з географічною прив'язкою до місцевості в окремих ІТС. Тому при формуванні використовується система (ІТС «Гарт-12»).

На сьогоднішній день вона використовується лише для висвітлення поверхневої ситуації в Азово-Чорноморському акваторії, тоді як існує також реальна можливість отримання інформації про поточне положення суден, порт відправлення та порт призначення, тип вантажу та інші додаткові дані. Коли вибрано певне судно, система поверхневого освітлення (SBS) відображає попередні точки свого маршруту, отримані в певні дискретні моменти часу, і маршрут візуалізується за допомогою лінійного наближення (існуючі точки з'єднані відрізками прямої лінії) [1].

Варто зазначити, що при аналізі інформації, відображеної в СВНО, виникають певні труднощі при виявленні потенційних загроз, пов'язаних із

порушенням прикордонного законодавства, а саме проблема виникає у разі відхилення судна від контрольного маршруту, що в деяких випадках може становити порушення правил прикордонного режиму. Беручи до уваги цю проблему, можна сформулювати нагальне завдання, а саме, аналіз та характеристика деяких додаткових ознак, які можуть свідчити про вказане порушення [44].

Ідентифікація цих ознак є досить складним завданням, оскільки воно безпосередньо пов'язане з необхідністю обробки значних обсягів даних про поверхневу ситуацію в ІТС "Гарт-12", що, в свою чергу, викликає певну складність аналізу ризиків. Тому для вирішення цієї проблеми запропонований підхід, пов'язаний зі створенням відповідної автоматизованої системи обробки даних, яка повинна містити процедури кластеризації маршрутів, встановлення опорних маршрутів для суднового руху в ІТС, виявлення аномалій суднового руху та оцінка ризиків. Зрозуміло, що ефективність безпосередньо залежить від ефективності функціонування цієї системи.

Водночас, незважаючи на існування підходів до комплексної оцінки ефективності функціонування СВУ, загалом, сьогодні питання оцінки ефективності автоматизованої системи обробки даних залишається невирішеним. Оскільки остання є інтелектуальною складовою ІТС, питання оцінки її ефективності є особливо актуальним завданням, яке потребує вирішення.

Таким чином, з аналізу випливає, що основним положенням, яке визначає ефективність автоматизованої системи обробки даних, є метод кластеризації маршруту суден. Цей метод виступає інтелектуальною основою для можливості виявлення ознак порушення прикордонного законодавства, є вирішальним впливом на ефективність досліджуваної автоматизованої системи за допомогою надійного результату, отриманого саме за допомогою цього методу.

Роботи в яких розглядаються принципи і взагалі ситуація із надводною обстановкою, контролювання розходження суден [25, 26 27] наведені у додатку.

Згідно з вищезазначеними фактами, виникає питання про необхідність оцінки надійності результатів, які формуються методом кластеризації маршрутів

кораблів. Слід підкреслити важливість вдосконалення цього методу завдяки використанню різних методів формування та визначення еталонного маршруту судна, що дозволить на основі цих результатів побудувати кіберфізичну систему, де це завдання буде впроваджено.

## 1.2 Існуючі методи виявлення просторових аномалій руху судна

Усі заходи щодо формування та розвитку наукових знань здійснюються за допомогою словесних засобів, підходів, методів. Роль методів у науковому пізнанні дуже важлива. Це було помічено давно. Навіть англійський філософ Френсіс Бекон, наголошуючи на величезній важливості методу, порівняв його з ліхтарем, який освітлює шлях мандрівника в темряві. Отже, є хороший метод - будуть вагомими результати.

Метод у найзагальнішому розумінні - це певний спосіб вивчення проблеми чи завдання, тобто метод - це система правил, принципів і методів підходу до вивчення процесів, явищ та мислення або практична перетворююча діяльність людини.

У цьому пункті розглядаються існуючі методи виявлення аномалій суднового руху.

Метод кластеризації - метод обробки даних, базується на групуванні даних, подібних між собою.

Головна мета. Кластерний аналіз термінів (вперше представлений Тріоном, 1939) складається з набору різних алгоритмів класифікації. Типове запитання, яке задають дослідники у багатьох галузях, полягає в тому, як організувати спостережувані дані у візуальні структури, тобто розширити таксономію. Наприклад, біологи прагнуть розбити тварин на різні види, щоб змістовно описати відмінності між ними [50]. Відповідно до сучасної системи, прийнятої в біології, людина належить до приматів, ссавців, амніотів, хребетних і тварин. У цій класифікації, чим вищий рівень агрегації, тим менша подібність між членами відповідного класу. Людина більше схожий на інших приматів (тобто мавп), ніж

на «віддалених» представників сімейства ссавців (наприклад, собак) тощо. Загальні методи кластеризації будуть розглянуті нижче.

Перевірка статистичної значимості. Попередня інформація стосується алгоритмів кластеризації, але нічого не згадує про тестування статистичної значимості. Насправді кластерний аналіз - це не стільки звичайний статистичний метод, скільки "набір" різних алгоритмів "розподілу об'єктів у кластери". Існує точка зору на те, що, на відміну від багатьох інших статистичних процедур, методи кластерного аналізу використовуються в більшості випадків, коли у вас немає апріорних гіпотез щодо класів, але вони все ще перебувають на етапі опису досліджень. Слід розуміти, що кластерний аналіз визначає "найбільш важливе можливе рішення". Тому тестування статистичної значущості насправді тут не застосовується, навіть у тих випадках, коли відомі рівні  $p$  (як, наприклад, у методі  $K$ -значень).

Сфери використання. Метод кластеризації застосовується в самих різних сферах. Наприклад, у галузі медицини кластеризація захворювань, лікування хвороб або симптоми захворювань призводить до широко використовуваної систематики. У галузі психіатрії правильний діагноз скупчень таких симптомів, як параноя, шизофренія тощо, є вирішальним для успішної терапії. В археології, використовуючи кластерний аналіз, дослідники намагаються встановити таксономії кам'яних знарядь праці, предметів поховання тощо. У маркетингових дослідженнях широко застосовуються кластерний аналіз. У нашому випадку ми використовуємо кластерний аналіз для виявлення просторових аномалій руху судна. Загалом, кожного разу, коли необхідно класифікувати "гори" інформації на відповідні групи для подальшої обробки, кластерний аналіз є корисним та ефективним.

Союз (кластеризація дерев). Загальна логіка. Наведений приклад пояснює мету алгоритму об'єднання (кластеризація дерев). Метою цього алгоритму є об'єднання об'єктів (наприклад, тварин) у досить великі скупчення, використовуючи певний ступінь подібності або відстані між об'єктами. Результатом є ієрархічне дерево.

Ієрархічне дерево. Проаналізуємо горизонтальну діаграму дерева. Діаграма починається з кожного об'єкта в класі (з лівого боку діаграми). А тепер уявімо, що поступово (дуже маленькими кроками) ви «послаблюєте» свій критерій щодо того, які об'єкти унікальні, а які ні. Іншими словами, знижений поріг відноситься до рішення об'єднати два або більше об'єктів в один кластер.

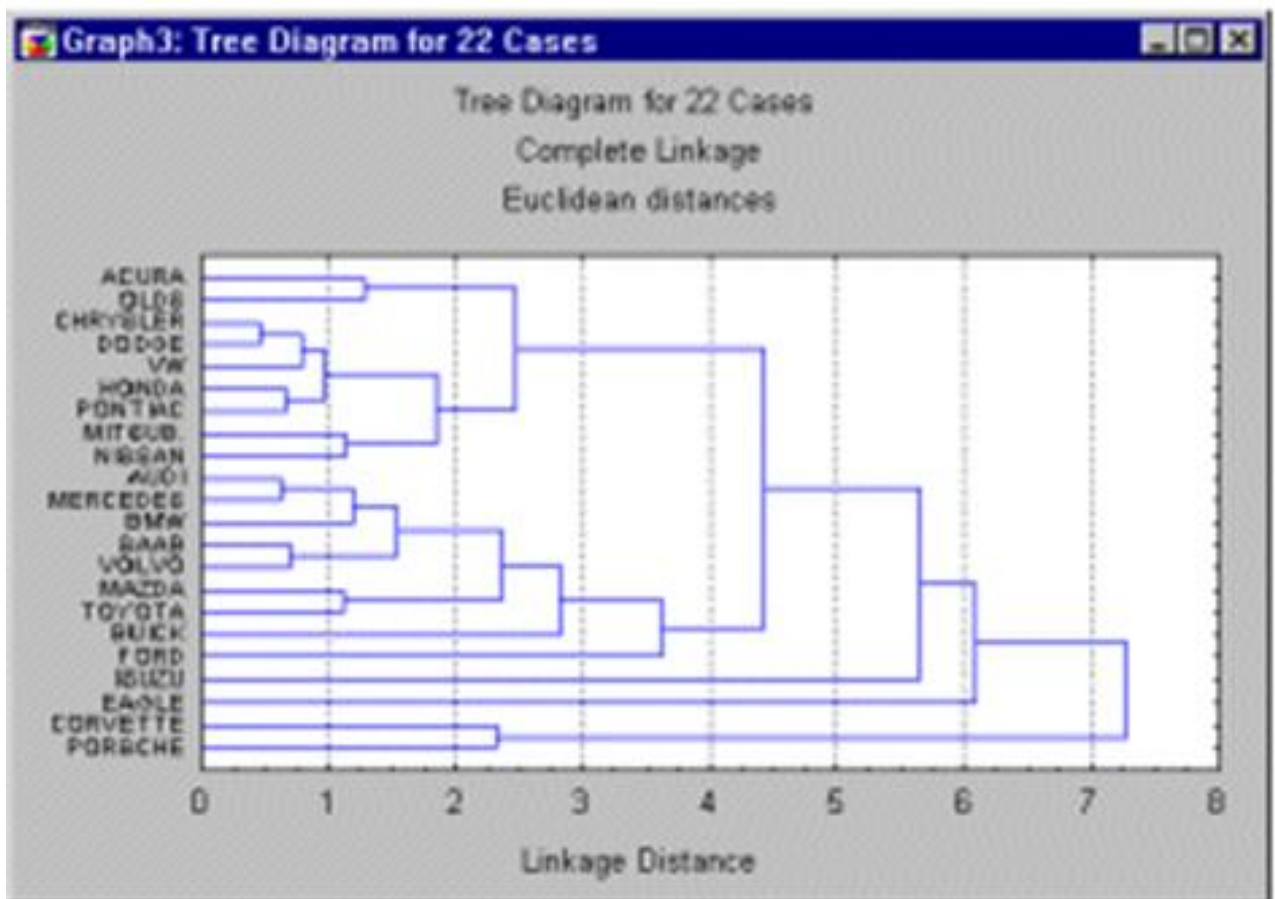


Рисунок 1.1- Горизонтальна діаграма дерева

В результаті ви зв'язуєте все більше і більше об'єктів разом і об'єднуєте все більше і більше кластерів все більшої кількості різних елементів. На останньому

кроці всі об'єкти об'єднані разом. У цих діаграмах горизонтальні осі представляють відстань з'єднання (у вертикальних діаграмах дерева вертикальні осі представляють відстань з'єднання). Отже, для кожного вузла на графіку (де утворюється новий кластер) ви можете бачити відстань, на яку відповідні елементи пов'язані в новий єдиний кластер. Коли дані мають чітку "структуру" з точки зору кластерів об'єктів, подібних один до одного, тоді ця структура, швидше за все, повинна бути відображена в ієрархічному дереві різними гілками. В результаті успішного аналізу комбінуючим методом стає можливим ідентифікувати кластери (гілки) та інтерпретувати їх.

Метод кластеризації дерев використовується при формуванні різнорідних кластерів або відстаней між об'єктами. Ці відстані можна визначити в одновимірному або багатовимірному просторі. Наприклад, якщо ви хочете згрупувати типи їжі в кафе, ви можете взяти до уваги кількість калорій, які вона містить, ціну, суб'єктивний смак тощо. Найбільш прямий спосіб обчислення відстані між об'єктами в багатовимірному просторі - це обчислення Евклідові відстані. Якщо ви перебуваєте у дво- чи тривимірному просторі, тоді ця міра є реальною геометричною відстанню між об'єктами в просторі (як ніби відстань між об'єктами вимірювали рулеткою). Однак алгоритм об'єднання не "байдужий", чи справжні "подані" відстані для цього чи інша похідна міра відстані.

Евклідова відстань. Найпоширеніший тип відстані. Це просто геометрична відстань у багатовимірному просторі. Евклідова відстань (і її квадрат) обчислюється на основі вихідних даних, а не стандартизованих даних. Це звичайний спосіб обчислення, який має певні переваги (наприклад, відстань між двома об'єктами не змінюється, коли в аналіз вводиться новий об'єкт, що може виявитись відхиленням). Однак на відстань можуть сильно впливати різниці між осями, за координатами яких ці відстані обчислюються.

Квадрат відстані Евкліда. Іноді вам може знадобитися звести в квадрат стандартну евклідову відстань, щоб надати більшу вагу об'єктам, які віддалені один від одного.

Відстань міських кварталів (відстань Манхеттески). Ця відстань є просто середнім значенням різниці координат. У більшості випадків ця міра відстані призводить до тих самих результатів, що і для звичайної евклідової відстані. Однак для цієї події вплив окремих великих відмінностей зменшується, оскільки вони не мають квадрата.

Відстань Чебишева. Ця відстань може бути корисною, якщо вам потрібно визначити два об'єкти як "різні", якщо вони відрізняються лише однією координатою (будь-яким одним виміром).

Степенева відстань. Іноді доводиться покроково піднімати або зменшувати вагу яка пов'язана з розміром, якому супровідні об'єкти дуже різні. Цього можна досягти, використовуючи дистанцію закону степенів.

Відсоток розбіжностей. Цей прийом використовується, коли дані категоричні.

Правила асоціації або посилення. На першому етапі, коли кожен об'єкт є окремим кластером, відстань між цими об'єктами визначається певним значенням. Однак коли кілька об'єктів пов'язані між собою, виникає питання, як слід визначати відстань між скупченнями? Іншими словами, вам потрібно правило об'єднання або зв'язку для двох кластерів. Тут є можливості: наприклад, можна зв'язати два кластери разом, коли будь-які два об'єкти в двох кластерах знаходяться ближче один до одного, ніж відповідна відстань посилення. Іншими словами, "правило найближчого сусіда" використовується для визначення відстані між кластерами. Це правило будує "волокнисті" скупчення, тобто скупчення, "зв'язані між собою" лише окремими елементами, які, як виявляється, знаходяться ближче один до одного.

Метод найближчого сусіда. / Як описано вище, у цьому методі відстань між двома скупченнями визначається відстанню між двома найближчими об'єктами у різних

Скупчення. Це правило полягає у об'єднанні об'єктів, щоб утворити кластери, і результуючі кластери, як правило, становлять довгі «ланцюжки».

Метод найвіддаленіших сусідів.

У цьому методі відстань між скупченнями визначається найбільшою відстанню між будь-якими двома ознаками в різних скупченнях (тобто «найдалшими сусідами»). Якщо скупчення мають «ланцюговий» тип, то цей метод не застосовується.

Середнє незважене парне. У цьому методі відстань між двома різними скупченнями обчислюється як середня відстань між усіма парами об'єктів у них. Метод ефективний, коли об'єкти насправді утворюють різні поля, але він однаково добре працює у випадку кластерних ланцюгів.

Середньозважене парне. Метод ідентичний методу незваженого парного середнього, за винятком того, що розмір відповідних кластерів використовується як ваговий коефіцієнт при розрахунках. Отже, запропонований метод слід застосовувати, коли передбачаються неоднакові розміри кластера.

Метод незваженого центру. У цьому методі відстань між двома скупченнями визначається як відстань між їх центрами ваги.

Метод зваженого центру. Ідентично попередньому, за винятком того, що в розрахунках використовуються ваги для обліку різниці між розмірами кластера. Отже, якщо є суттєві відмінності у розмірах кластерів, цей метод виявляється переважно попереднім.

Метод Уорда. Цей метод відрізняється від усіх інших методів, оскільки використовує аналіз дисперсійних методів для оцінки відстаней між кластерами. Метод мінімізує суму квадратів для будь-яких двох кластерів, які можна сформувати на кожному кроці. Детальніше можна знайти у Ward (1963). Загалом, метод досить ефективний, але він схильний створювати невеликі кластери.

Отже, кластерний аналіз можна визначити як процес розподілу набору об'єктів в окремі підмножини на основі їх подібності. Мета аналізу - отримати кластери, з одного боку, максимізувати мінливу мінливість кластера, а з іншого боку, мінімізувати внутрішню відстань кластера.

Іншим методом виявлення просторових аномалій руху судна є метод супутникової геодезії. Розміщення судна є важливою складовою безпеки судноплавства. Позиціонування судна здійснюється за даними супутникової

геодезії. Ефективність методу супутникової геодезії залежить від ряду помилок, включаючи невідповідність положення географічного центру Землі та центру її маси. На рис. 1.2.2 показана схема геоцентричної системи WGS-84, згідно з якою центр відліку є географічним центром планети.

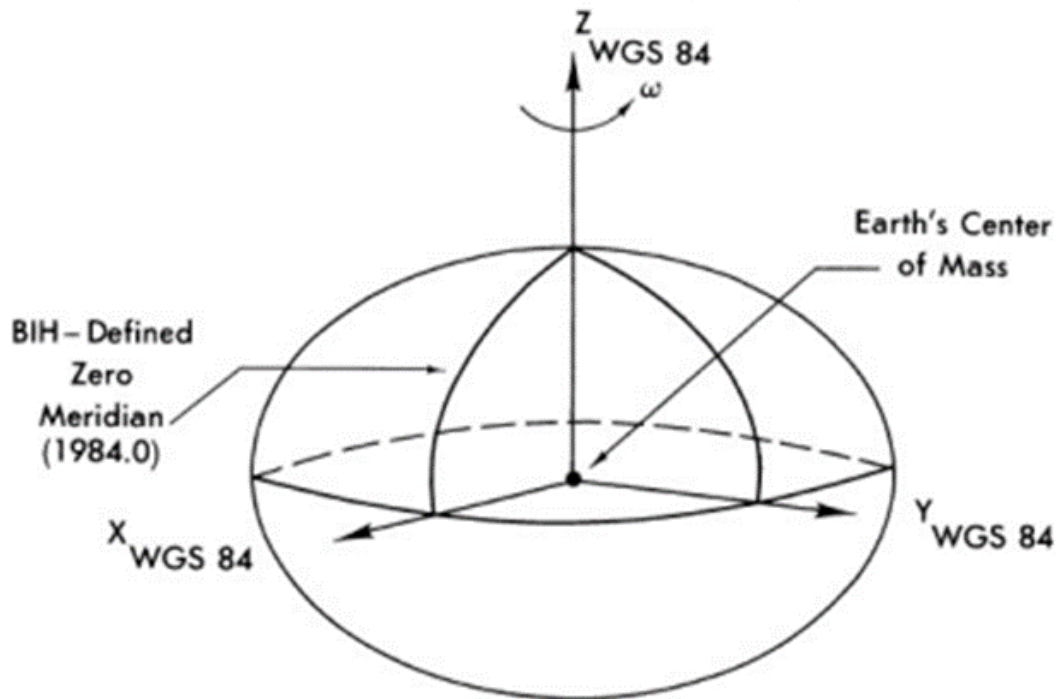


Рисунок 1.2 - Топографія морської поверхні отримана в геоцентричній системі координат WGS-84



Рисунок 1.3- Перерахування відносно еталонної поверхні (тобто поверхні геоїду)

Методи супутникової геодезії засновані на геодезичних супутниках як геодезичних орієнтирах, точність розташування яких залежить від точності визначення орбітальних параметрів супутників. Геодезичні супутники обертаються навколо центру маси Землі, що може не збігатися з її географічним центром, є причиною помилок у визначенні координат у Всесвітній геодезичній системі (WGS-84). Слід зазначити, що існують і інші процеси, що формують орбітальні помилки. Як приклад, орбітальні помилки створюються "новим глобальним режимом геодформацій Землі", який має річну циклічність. Важливо відзначити, що різниця між формою морської поверхні та формою геоїда може бути пов'язана з ефектом обертання рідинної оболонки Землі. На рис. 4 показано математичне моделювання "мілководдя" на кулі, що вказує на можливість існування власної форми рідинної оболонки на кулі, що обертається, що також може створити помилку позиціонування.

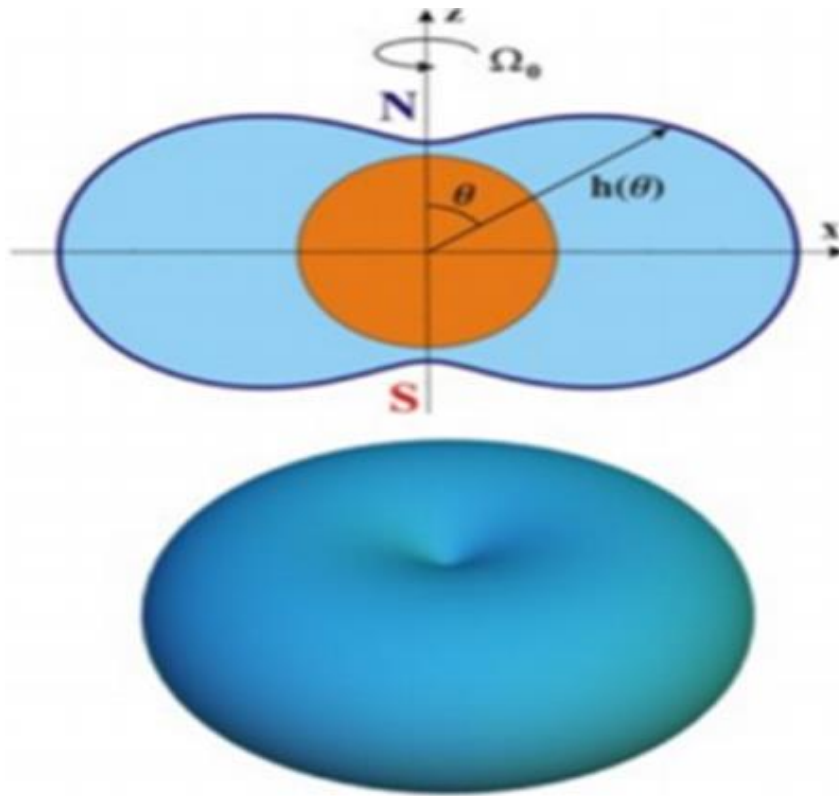


Рисунок 1.4-Математичне моделювання "мілкої" води

Схема природної форми рідинної оболонки в геоцентричній системі координат на кулі, обертається.

Точність методів супутникової геодезії залежить від ряду помилок, включаючи невизначеність орбітальних характеристик супутників під впливом гравітаційного поля планет, невизначеність орбітальних характеристик геодезичних супутників в умовах різниці в положенні географічного центру Землі та центру її мас тощо. Саме тому, що цей метод залежить від багатьох змін, він недостатньо точний і не надто придатний для виявлення аномалій та контролю руху судна. Більш детально даний метод був розглянутий авторами роботи [35].

Кожен із розглянутих вище методів має свої переваги та недоліки. Залежно від обраного методу будується система для його реалізації. Але в цій роботі буде розглянуто метод кластеризації.

### 1.3 Існуючі засоби виявлення просторових аномалій руху судна

Будь-яка діяльність вимагає засобів, за допомогою яких вона буде здійснюватися. Засобами можуть бути як очищення програмного забезпечення, так і конкретні інструменти. Загалом визначення засобу звучить як набір прийомів, інструментів для вирішення, досягнення чогось. Тому наведемо кілька прикладів інструментів для виявлення та контролю руху суден.

Система моніторингу поверхневої ситуації та ZNO. Система зберігається з апаратного та програмного забезпечення обласного центру (АПЗ РЦ), розташованого в адміністративному корпусі філії.

Система повинна підвищувати рівень безпеки судноплавства, ефективність навігації та експлуатації систем управління судновим рухом, охорону навколишнього середовища та забезпечувати виконання таких функцій, як:

1. Уникнення зіткнень у режимі «корабель-судно».
2. Оримання компетентними прибережними службами інформації про судно та вантаж.
3. Управління судновим рухом у режимі судно-берег.
4. Моніторинг та нагляд за кораблями та участь у пошуково-рятувальних операціях (SAR).

Система AIS включає:

1. УКВ-передавач.
2. УКХ приймач.
3. GPS приймач.
4. Модулятор / демодулятор.
5. Кнтролер на основі мікропроцесора.
6. Пристрої введення / виведення інформації про елемент управління.

Робота AIS заснована на прийомі та передачі повідомлень через УКВ-радіо. Оскільки передавач AIS працює на довжинах хвиль більше, ніж радари, це дозволяє обмінюватися інформацією не тільки на прямих відстанях, але і на

місцевості з перешкодами у вигляді дрібних предметів, а також у несприятливих погодних умовах.

Повідомлення AIS можуть містити:

- 1) ідентифікаційна інформація про об'єкт;
- 2) інформація про стан об'єкта отримується автоматично від елементів управління об'єкта (в тому числі від деяких радіонавігаційних пристроїв);
- 3) інформація про географічні та часові координати, отримані від Глобальної навігаційної супутникової системи;
- 4) інформація, введена вручну персоналом, що обслуговує заклад.

Він також передбачений для передачі додаткової текстової інформації між терміналами AIS (пейджингове повідомлення), яка може передаватися як на адресу всіх терміналів в діапазоні, так і на будь-який конкретний термінал.

Робота кожної станції AIS (мобільної чи базової) чітко синхронізована за часом UTC з помилкою не більше 10 мкс від вбудованого приймача глобальної супутникової системи ГЛОНАСС / GPS. Для передачі інформації використовуються постійно повторювані кадри тривалістю 1 хв, які розподіляються у 2250 часових інтервалах (інтервалах) тривалістю 26,67 мс. Текстові повідомлення використовують 6-розрядні коди ASCII. Структура повідомлень в системі AIS за типом інформації поділяється на статичну, динамічну та іншу інформацію.



Рисунок 1.5 - Технічні засоби автоматизованої базової системи для моніторингу поверхневої ситуації



Рисунок 1.6 - Апаратне та програмне забезпечення регіонального центру

Служба регулювання руху суден (СРСР) є одним з основних підрозділів відділення Дельта-Пілот Державного підприємства «Адміністрація морських портів України», яке забезпечує судноплавство в Азово-Чорноморському басейні України та на українській частині Дунаю.

СРПС включає 10 центрів та постів для регулювання руху суден (CRDS і PRDS), 7 автоматизованих радіолокаційних постів (ARLP), 3 радіотехнічні пости (RTPZ) та оперативний координаційний центр (ОКТ). Прилади SRDS оснащені радіолокаційними та метеорологічними станціями, автоматизованими системами

ідентифікації, засобами зв'язку та спеціальним програмним та математичним обладнанням; вони працюють цілодобово в безперервному і безперебійному режимі.

Основні функції об'єктів SRDS:

- ідентифікація суден на підходах до зони покриття SRDS та встановлення зв'язку з ними;
- надання судам навігаційної, гідрографічної та гідрометеорологічної інформації, інформації про присутність суден на їх маршрутах, навігаційна допомога у складних метеорологічних та навігаційних умовах;
- організація руху та контроль руху суден за відповідними планами та графіками;
- надання судів суднам щодо черги руху, часу старту, маршруту, їх швидкості, місць стоянки;
- попередження суден про порушення правил судноплавства за маршрутом їх руху;
- попередження суден на випадок надзвичайних ситуацій та надання їм рекомендацій щодо запобігання зіткненням з іншими кораблями, а також небезпечним відхиленням від встановлених шляхів руху;
- збір, обробка, реєстрація та архівування інформації про судна (їх ім'я, власник судна, вантаж, час перебування в портах та на якорних пунктах тощо).

Це одна з небагатьох приватних чи державних систем, що підтримують порядок на морі. Однак кожна держава має власні військові засоби для контролю судноплавства та забезпечення прикордонної безпеки.

Ось кілька прикладів цього.

Інтегрована система тактичного спостереження ВМС США (ITSS)

Складається з наземних, авіаційних, космічних та корабельних зв'язків. Він отримує інформацію від різних інструментів моніторингу та використовує банк даних для всіх компонентів, що входять до системи.

Американська система була створена за принципом зонування, в кожному регіональному центрі зосереджена інформація про власну зону відповідальності.

Інформаційні центри оснащені засобами для збору, обробки та розповсюдження інформації в районі діяльності комплексу. Постійний моніторинг, контроль та отримання інформації про морську ситуацію в глобальному масштабі, діяльність сил потенційного ворога та своєчасне інформування командних пунктів відповідних сил реагування є одним з найважливіших завдань Міністерства оборони США .

Система моніторингу та контролю шведського флоту 9CSI 600 Wall

Система дозволяє здійснювати спостереження, автоматизований обмін інформацією та управління силами в зоні дії. Він має два обласні центри, які, в свою чергу, мають спостережні пункти. Він відрізняється високою швидкістю передачі даних, високим рівнем автоматизації процесів виявлення, обробки, передачі інформації та призначення цілей. Система дозволяє одночасно відстежувати до 400 цілей, з яких 200 знаходиться в автоматичному режимі. Створений на регіональній основі і охоплює все узбережжя країни.

Система SCOMAR (Sistemul Integrat de Observare, Supraveghere and Control La MAREa Neagră). SCOMAR - це складна система, яка є результатом інтеграції різних підсистем. Відрізняється постійною цілодобовою готовністю, що забезпечує тактичне зображення поверхневої, наземної та повітряної обстановки на морському кордоні в Чорному морі; румунський сектор східного Дунаю та дельти Дунаю в морській зоні.

Система SCOMAR складається з 5 компонентів, а саме: наземний, надводний, повітряного спостереження, командування та реагування.

Під час роботи системи розвідувальна інформація, отримана наземними (NKS), морськими (ISS), повітряно-десантними (PPS) компонентами спостереження, передається до командного компонента, де обробляється та аналізується. Надалі командний центр може отримувати вказівки щодо уточнення областей особливої уваги або, у разі кризової ситуації, наказу використовувати компонент реагування.

Оперативно-функціональним компонентом системи SCOMAR є оглядова вежа або бортовий комплекс. Він включає необхідну кількість датчиків,

радіостанцію, ідентифікаційну станцію, локальну мережу зв'язку, сервер та консоль оператора. Відповідно до адміністративного підпорядкування та функціонального призначення системи SCOMAR вона складається з двох підсистем:

Прикордонна поліція та військово-морські сили. Підсистема системи прикордонної поліції SCOMAR, яка належить Міністерству адміністрації та внутрішніх справ (MAVS) Румунії, включає такі елементи: Радіолокатор ближнього поля до 30 морських миль (NKS), оптоелектронне обладнання для розвідки та спостереження (NKS) компонент реагування патрульних літаків (PPS) патрульних кораблів та катерів (ISS); цифрові канали зв'язку.

Загалом, в інтересах наземної складової спостереження за підсистемою прикордонної поліції було побудовано 8 оглядових веж висотою 20 м, оснащених 8 РЛС діапазону та 8 оптоелектронними системами Raytheon. у відкритому морі на відстані до 6 км в межах від 7,5 до 13 мкм. Загалом, система дозволяє спостерігати за об'єктом за допомогою телевізора та в інфрачервоному діапазоні за азимутом на 360°, з висотою від -35° до 60°.

Як підводний компонент для проведення гідроакустичної розвідки в системі SCOMAR використовується комплекс Cerberus 360 компанії QnetiQ, який забезпечує виявлення підводних об'єктів на відстані до 700 м. 10 патрульних кораблів були включені до надводного компоненту спостереження за підсистемою прикордонної поліції системи SCOMAR. В даний час повітряна складова наземного спостереження представлена вертольотами спеціального авіаційного підрозділу прикордонної поліції МАВС. У майбутньому він планує придбати у Raytheon патрульний літак КА 350ER.

Компонент реагування підсистеми прикордонної поліції MAVS включає три компоненти:

- морський-31 корабель;
- наземний - 88 транспортних засобів;
- повітряні - вертольоти спеціального авіаційного підрозділу МАВС.

Система зв'язку між функціональним та оперативним компонентами підсистеми прикордонної поліції була розроблена та впроваджена італійською компанією SALEX COMMUNICATION SA. Основою стало розгортання еталонної системи радіорелейного зв'язку між компонентами підсистеми прикордонної поліції зі швидкістю передачі інформації 4 МБ / с. З метою реалізації командного компоненту підсистеми прикордонної поліції MAVS на території відділу прикордонної поліції в Констанці було побудовано центр управління (КМП). Командно-диспетчерський центр обладнаний засобами для обробки та відображення розвідувальної інформації від компонентів спостереження та необхідними засобами зв'язку. Підсистема військово-морських сил системи SCOMAR, що перебуває під юрисдикцією Міністерства національної оборони Румунії, включає радар для середньої та дальньої зони виявлення:

- засоби радіорозвідки та пошуку напрямків (НКС, МКС);
- засоби ідентифікації (ідентифікатори "друг або ворог") (NKS, ISS);
- гідроакустичні засоби розвідки (НКС, МКС);
- засоби розвідки RCB (НКС, МКС);
- вертольоти IAR-330 PUMA-NAVAL (PPS).

Завдяки всім цим системам контроль над поверхневою ситуацією здійснюється в тому стані, де він використовується. Кожна з цих систем по-своєму відрізняється і по-своєму реалізує покладені на неї обов'язки. Але всі вони будуть підтримувати безпеку та територіальну цілісність своїх держав, де вони перебувають у провінціях, і завдяки їм забезпечується безпека на морській території.

## 1.4 Висновки

Одним із пріоритетів держави є її безпека. А безпека на морі - одна із складових. У цьому розділі були розглянуті приклади доступних методів та інструментів для виявлення аномалій та контролю руху судна, демонструючи їх основні функції, складові та переваги з недоліками. Кожен спосіб і кожен засіб по-своєму унікальні. Тому є такий великий вибір для військових та людей, причетних до цього. У цій роботі буде розглянуто метод кластеризації, який реалізований на основі системи "Гарт-12", яка буде розглянута в наступних главах.

## 2 МЕТОД КЛАСТЕРИЗАЦІЇ МАРШРУТІВ РУХУ СУДЕН

### 2.1 Метод кластеризації маршрутів руху суден

При аналізі даних виявлення аномалії відноситься до пошуку та ідентифікації елементів, подій чи спостережень, які не відповідають очікуваній поведінці чи іншим елементам набору даних. У цьому контексті, аналізуючи дані в ІТС Морської гвардії, необхідно вирішити ряд конкретних проблем:

- формування набору типових маршрутів на основі аналізу даних у базі даних "Гарт-12";
- проведення класифікації нових маршрутів;
- виявлення аномалій та оцінка ризику на їх основі.

Для кожного маршруту в базі даних "Гарт-12" міститься інформація про порт відправлення, порт призначення і тип вантажу. Ці дані можна використовувати для попередньої класифікації маршруту. Потім, після подальшої просторової класифікації, ідентичність цих результатів може бути використана для оцінки ризику. Однак з часом типові маршрути можуть змінюватися через появу нових проміжних зупинок. Отже, згідно з попередньою класифікацією, одному просторовому шляху може відповідати кілька просторових маршрутів. Належність досліджуваного шляху до одного з них не є підставою для зростання ризику. Під час руху кожного судна його координати записуються в базу даних Gart-12 в окремі дискретні моменти часу. Позначимо набір цих координат для  $i$ -го судна упорядкованою множиною (кортежем)  $M_i = \{(x, y), \dots, (x_{k_i}, y_{k_i})\} 1 + 1$ , де  $k_i$  - потужність множини  $M_i$ . Визначення аномалій дорожнього руху та класифікація маршрутів вимагає встановлення ступеня їх подібності, який можна визначити на основі введення відповідної метрики. Відстань між двома маршрутами  $M_1$  і  $M_2$  позначається  $R(M_1, M_2)$ . Цю відстань можна визначити на основі стандартної метрики між окремими точками множин  $M_i$ , які представляють. Однак відмінності в потужностях цих множин ускладнюють таке визначення метрики. Основна проблема такого підходу полягає в тому, що окремі точки представлення маршруту отримуються в різний час. Навіть коли два

маршрути збігаються, відстань між окремими дискретними точками, що описують, може бути значною.

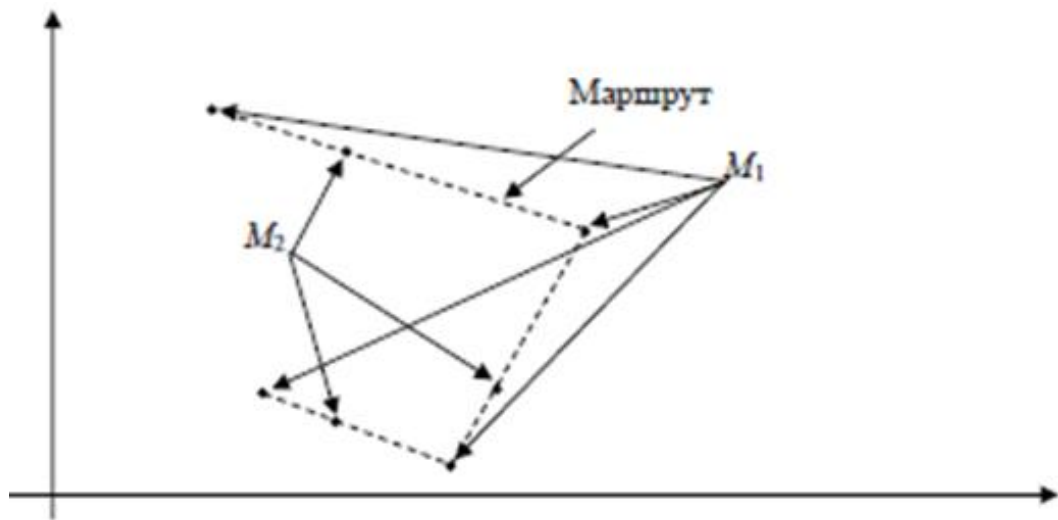


Рисунок 2.1- Відмінності при представленні маршруту, обумовлені різницею відліків у часі

Відповідно до рисунку перший маршрут представлений набором, що містить чотири елементи. Другий маршрут на рис. Описується набором з трьох елементів. Всі пункти цих маршрутів лежать на одній пунктирній лінії. Оскільки ці показання були зроблені в різні моменти часу, відстань між ними значна. У зв'язку з цим неправильно будувати метрику на основі відстаней між окремими дискретними вибірками. Враховуючи, згаданий вище, для знаходження відстані між двома маршрутами, представленими дискретно упорядкованими наборами зразків, пропонується відновити один із цих маршрутів, використовуючи наближення між кожною парою сусідніх доступних точок. Нехай:  $M_1$  - набір, що описує маршрут, що підлягає відновленню,  $f_k(x)$  - це функція, яка апроксимує перший маршрут між точками  $(x_k, y_k)$  та  $(x_k + 1, y_k + 1)$ .

Потім, щоб визначити метрику з точки зору геометричної близькості маршрутів, для будь-якого посилання описує інший маршрут, пропонується знайти найближчу точку апроксимованого першого маршруту (рис. 2.1).

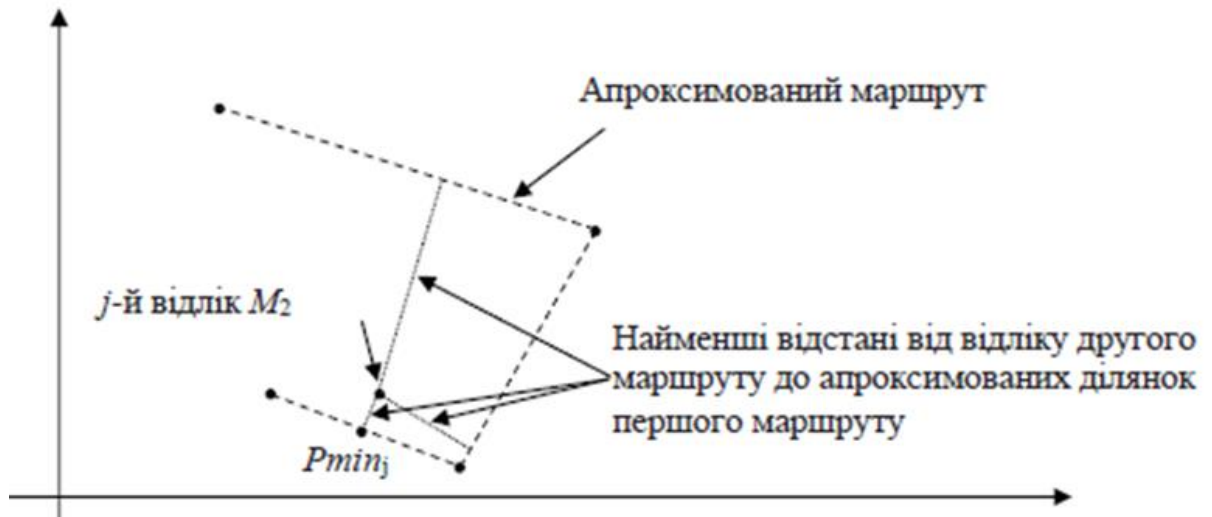


Рисунок 2.2- Використання апроксимації маршруту для визначення метрики

На рис. 2.2 показано наближення з використанням лінійного співвідношення. Однак у загальному випадку можна вибрати більш складну функцію. З прикладу (рис. 2.2) досліджувана  $j$ -та точка цього другого маршруту відповідає точці першого маршруту  $P_{minj}$ , мінімальна відстань від якої до трьох наближених ділянок виявилася найменшою. Ця відстань буде позначатися  $R_{minj}$ . Такі відстані пропонується взяти за основу при визначенні метрики  $R(M1, M2)$ , яку ми визначаємо як(один):

$$R(x) = \sqrt{(x - x_1)^2 + (y_1 - f_1(x))^2}. \quad (2.1)$$

Звичайно, головним проблемним питанням при розрахунку є обчислення значень  $R_{minj}$ , для яких необхідно визначити точки  $P_{minj}$ . Розглянемо можливий підхід до вирішення цієї проблеми. Для  $i$ -го наближеного відрізка першого маршруту передбачена точка  $(x, y)$ , для якої найменша відстань до одного з дискретних зразків другого маршруту  $(x_1, y_1)$ .

Тому, щоб знайти мінімальне значення, необхідно прирівняти похідну радикального виразу до нуля і розв'язати рівняння

Якщо рівняння матиме кілька розв'язків, необхідно визначити остаточне рішення за критерієм мінімальності.

За цих умов вона матиме вигляд функції з єдиним мінімумом (рис.2.3). У цьому випадку рішення (3) має бути однорідним і відповідати необхідному мінімуму. Це дозволяє не перевіряти мінімум отриманого екстремуму.

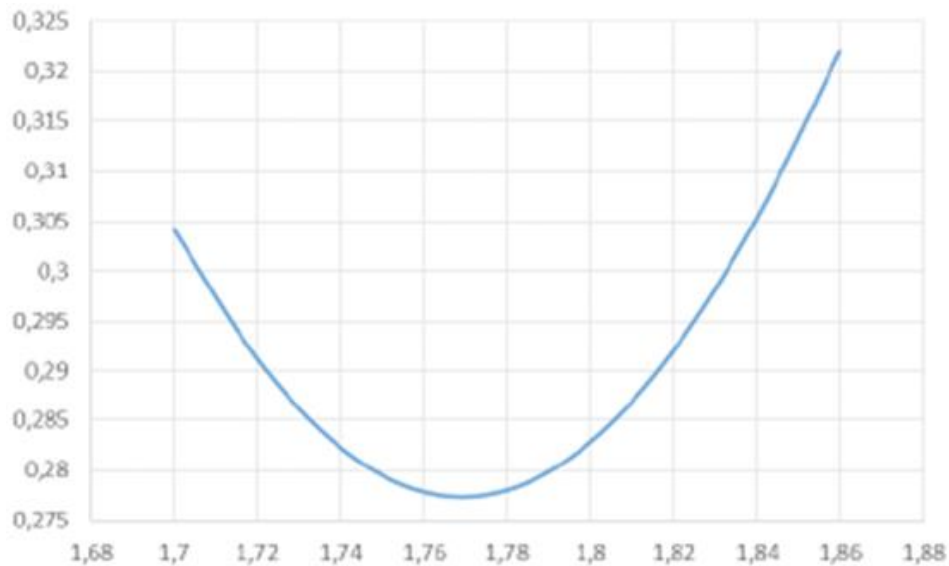


Рисунок 2.3- Залежність відстані при лінійній апроксимації

Беручи до уваги, можна отримати бажану точку  $(x, y)$ . Слід зазначити, що у випадку лінійного наближення вираз можна отримати на основі більш простих міркувань. Очевидно, що шукана точка буде відповідати точці перетину з перпендикулярною до неї прямою, що проходить через точку  $(x_1, y_1)$ .

Точка перетину визначається із системи:

$$\begin{cases} y = k_i * x + b_i \\ y - y^1 = -\frac{1}{k_i}(x - x_1) \end{cases} \quad (2.2)$$

Отже, це дає можливість визначити мінімальні відстані від кожної  $j$ -ї точки другого маршруту до різкого, який наближається до першого маршруту. Найменша з цих відстаней  $R_{minj}$  використовується у виразі для обчислення остаточної метрики для встановлення ступеня подібності між маршрутами.

Оптимізація навігаційного обладнання

Сучасні засоби навігації поділяються на пряму і зворотну системи. Системи прямого типу - це низка природних та штучних навігаційних орієнтирів, завдяки яким можна виміряти параметри навігації з корабля та за допомогою суднових засобів визначити місце його спостереження. Системи прямого типу - це також спеціально організовані локальні системи.

Радіонавігаційні локальні системи зворотного типу самостійно визначають точні координати судна і передають їх. Багатопозиційна радіолокаційна система зворотного типу забезпечує точність позиціонування судна, порівнянну з точністю супутникових навігаційних систем. Якщо точність навігації судна по заданому маршруту нижча за діючі нормативні вимоги, то виникає необхідність підвищити його, встановивши систему навігаційного обладнання, тобто визначити його тип і структуру, що забезпечують необхідну точність навігації .

Доступні системи прямого та зворотного типу, одного з трьох класів: стандартні системи, доповнені штучними навігаційними опорними точками, кореляційні радіолокаційні системи та багатопозиційні радіолокаційні системи зворотного типу.

В усіх трьох класах головним обмежуючим фактором є точність контролю положення судна на заданому маршруті, яка не повинна бути нижче необхідного стандартного показника точності.

Іншими словами, це завдання синтезу системи навігаційного обладнання з мінімальними витратами, яка повинна забезпечувати задану точність контролю положення судна в локальній зоні з необхідним рівнем надійності отримання високоточної інформації. Є багато можливих системних структур як альтернатив. Завдання оптимізації має такий аналітичний вираз:

$$\begin{aligned} C_0 &= \min C(G_{SO}), & D_r &\leq \overline{D_r}, \\ \min P &\geq \hat{P}, G_{SO} \in Mn_R. \end{aligned} \quad (2.3)$$

де  $C$  і  $C_0$  - це відповідно вартість і мінімальна вартість розглянутої системи.  
 $G$  - оптимальна структура системи;

$D_r$  - показник точності довільної точки  $(X, Y)$  області  $Q_r$ ;

$D_r$  - задане значення показника точності;

$P$  і  $P$  - це ймовірність отримання високоточної інформації та її бажаного значення відповідно.

В якості показника точності була обрана дисперсія модуля векторної похибки спостережуваного розташування судна  $D_r$ , яка є сумою дисперсій його складових  $D_x + D_y$ , тобто  $D_r = D_x + D_y$ .

За нормальним законом ймовірності розподілу помилок навігаційних параметрів, коли щільність розподілу має вигляд:

$$f(\xi) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left\{-\frac{\xi^2}{2\sigma_i^2}\right\}. \quad (2.4)$$

показник точності буде виглядати так:

$$D_r = \frac{1}{\Delta_r} \sum_{i=1}^n \sigma_i^2. \quad (2.5)$$

Де:

- $n$  – число ліній розташування;
- $i$  – середньоквадратичне відхилення похибки  $i$ ;

$$\Delta_r = [(\sum_{i=1}^n \cos^2 a_i)(\sum_{i=1}^n \sin^2 a_i) - (\sum_{i=1}^n \sin a_i \cos a_i)^2]. \quad (2.6)$$

- $a_i$  – напрямок градієнта  $i$ -й лінії положення.

При русі за заданим маршрутом для контролю положення судна візуальними методами можна практично використовувати не більше трьох орієнтирів. За допомогою однієї опорної точки положення судна можна отримати за опорою та відстанню опорної точки, а за наявності двох або трьох опорних точок для управління судном можна використовувати не більше трьох ліній

положення судна. При використанні двох рядків положення вираз для  $\Delta r$  набуває вигляду:

$$\Delta_r = (\cos^2 a_1 + \cos^2 a_2)(\sin^2 a_1 + \sin^2 a_2) = \sin^2(a_1 - a_2) = \sin^2 \beta_{12}. \quad (2.7)$$

а показник точності має вигляд:

$$D_r = \frac{\sigma_1^2 + \sigma_2^2}{\sin^2 \beta_{12}}. \quad (2.8)$$

При визначенні місця судна за трьома лініями положення величина  $\Delta r$  виражається як:

$$\Delta_r = (\cos^2 a_1 + \cos^2 a_2 + \cos^2 a_3)(\sin^2 a_1 + \sin^2 a_2 + \sin^2 a_3) - (\sin a_1 \cos a_1 + \sin a_2 \cos a_2 + \sin a_3 \cos a_3)^2. \quad (2.9)$$

Або

$$\Delta_r = \sin^2 \beta_{12} + \sin^2 \beta_{13} + \sin^2 \beta_{23}. \quad (2.10)$$

де  $\beta_{ij}$  – кути між градієнтами відповідних ліній розташування.

В такому випадку точність виражається так:

$$D_r = \frac{\sum_{j=1}^N \left( \frac{1}{\sigma_{Dj}^2} + \frac{1}{\sigma_{Pj}^2} \right)}{\frac{1}{2} \left( \sum_{i=1}^N \sum_{j=1}^N \left[ \left( \frac{1}{\sigma_{Dj}^2} + \sin^2 \beta_{ij} + \left( \frac{1}{\sigma_{Di}^2} + \frac{1}{\sigma_{Pi}^2} \right) \cos^2 \beta_{ij} \right] \right) \right)}. \quad (2.11)$$

Ось чому, маючи оцінку точності контролю положення судна, коли системи навігаційного обладнання належать до першого класу, тобто природних або штучних орієнтирів, значення показника точності обчислюється за формулами (1) або (2) (залежно від кількості та типу позиційних ліній). Маршрут судна

знаходиться в зоні навігації, яка може бути обладнана локальною зворотною радіолокаційною системою або локальною системою з використанням кореляційних принципів навігації.

Окрім конкретних технічних обмежень (наприклад, таких як дальність, надійність тощо), для класів систем навігаційного обладнання є обмеження щодо розміщення системних засобів, враховуючи площу їх можливих встановлення. Формалізація вищезазначеного обмеження зазнає значних труднощів, оскільки для кожного локального розміщення систем навігаційного обладнання необхідно сформулювати базу даних, що описує область, в якій можуть бути розміщені засоби. Крім того, така територія може бути сукупністю декількох окремих областей.

При використанні моделі для вибору оптимальної структури системи навігаційного обладнання розробник генерує оптимальну структуру системи навігаційного обладнання, а також оскільки оцінка допустимої площі плавання з аналізом розподілу значень показника точності плавання в ній покладається на розробника. Для кожного з трьох класів систем навігаційного обладнання береться мінімальна кількість її компонентів (наприклад, мінімальна кількість радарів станцій реверсної системи дорівнює двом). Основним обмеженням є вказане значення доктора. З урахуванням допустимих площ розміщення компонентів системи навігаційного обладнання, що забезпечує найбільшу точність у всіх точках обладнаної зони з заданим маршрутом плавання судна.

Далі, мінімальне значення індексу точності  $D_r$  на даному маршруті порівнюється із зазначеним значенням індексу точності  $D_r$ . Якщо обмеження точності не виконуються, тоді необхідно збільшити кількість компонентів системи на одиницю. Процедура повторюється до тих пір, поки не будуть виконані обмеження точності контролю положення судна розглянутою системою навігаційного обладнання. Ця процедура застосовується до кожного з трьох класів систем навігаційного обладнання. Це визначає собівартість витрат  $C_i$ . Очевидно, що перевага віддається варіанту, вартість реалізації якого найнижча. У цьому випадку критерій оптимальності досягає граничного значення, коли виконуються обмеження точності судна до зазначеного маршруту та обмеження розміщення

компонентів системи навігаційного обладнання в допустимих зонах. Для вирішення проблеми синтезу системи навігаційного обладнання для прибережних районів навігації кораблів необхідно розробити інформаційну систему з електронними картами, яка дозволить відбирати карти потрібної зони навігації з каталогу карт і визначити позиційні характеристики точності необхідного маршруту навігації.

## 2.2 Удосконалений метод кластеризації маршрутів суден

У базі даних "Гарт-12" інформація про рух суден зберігається у вигляді впорядкованої послідовності пунктів з координатами кораблів у певні моменти часу.

Ідею розташування кораблів у фіксований момент часу можна сформуванати, наприклад, використовуючи рис.2.1.

Набір координат точок, місце розташування  $i$ -го судна можна позначити упорядкованим набором (кортежем).

Слід зауважити, що потужності відповідних множин для різних суден, у загальному випадку, різні. Причому потужності цих множин можуть відрізнятися достатньо суттєво навіть у випадку, якщо судна рухаються по одному маршруту між заданими пунктами відправлення та призначення.

Траєкторії руху суден у вигляді неперервних кривих відсутні.

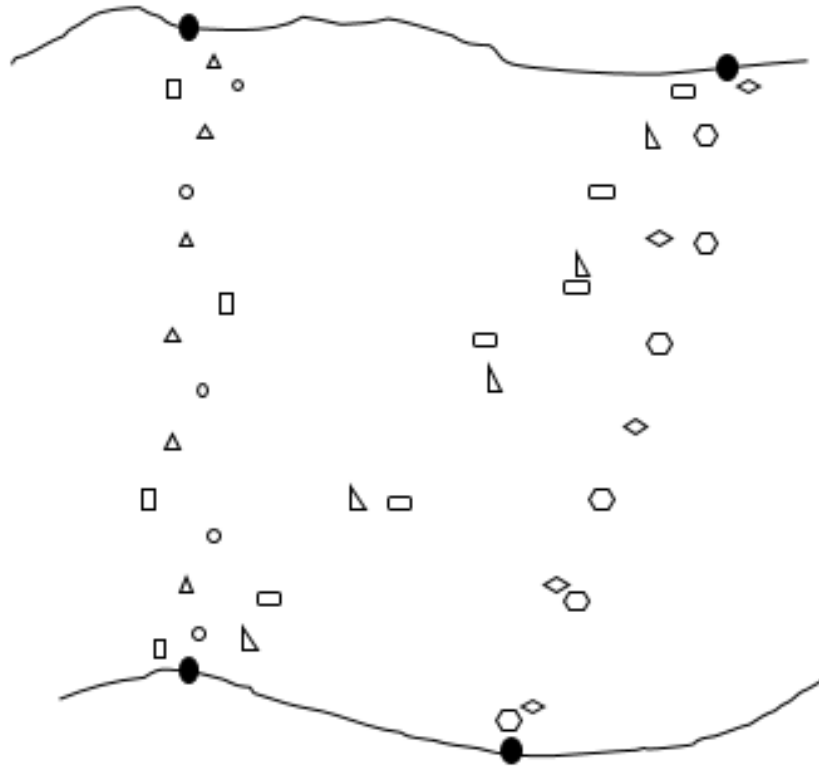


Рисунок 2.1 – Схематичне представлення даних щодо положення окремих суден у СВН.

Дослідження питання можливості порушення кораблем правил прикордонного режиму передбачає вирішення комплексу взаємопов'язаних завдань:

1. Встановлення подібності маршрутів довільних пар різних суден;
2. Створення групи маршрутів між визначеними пунктами відправлення та призначення (транспортний коридор між двома пунктами);
3. Визначення еталонного маршруту між двома точками в межах сформованого кластера;
4. Визначення належності траєкторії руху судна кластеру маршрутів між даними пунктами відправлення та призначення;
5. Оцінка ступеня відхилення траєкторії досліджуваного судна від еталонного маршруту.

У рамках цієї роботи аналізуються лише перші два завдання.

Завдання 1.

Нехай маршрути руху пари кораблів подаються у вигляді наборів типу (1).

Ступінь подібності маршрутів та (див. рис. 2) у роботі [1] запропоновано визначати за допомогою метрики .

Для розрахунку необхідно насамперед здійснити встановлення (відновлення, генерування) неперервного маршруту руху першого судна між заданими у дискретній множині точками з використанням лінійної апроксимації . Величина визначається, як середнє відхилення точок множини , які описують дискретний маршрут руху другого судна, від апроксимованого неперервного першого маршруту у вигляді:

$$R(M_1, M_2) = \frac{\sum_{j=1}^{k_2} R_{\min j}}{k_2} . \quad (2.12)$$

де  $R_{\min j}$  є найкоротшою відстанню від  $j$ -ї точки множини до встановленого першого маршруту, тобто до найближчого деякого  $i$ -го сегменту прямої, який з'єднує точки  $(x_i, y_i)$  і  $(x_{i+1}, y_{i+1})$  множини , а - кількість досліджуваних точок другого маршруту.

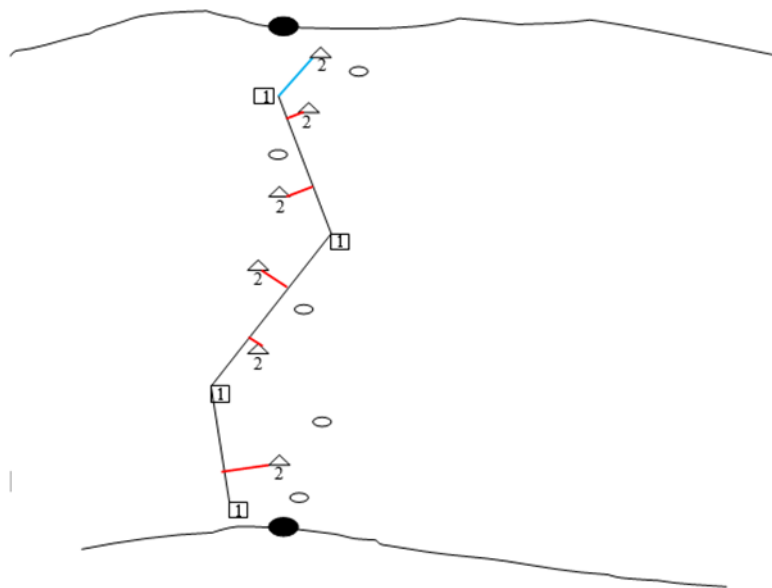


Рисунок 2.2 – Підхід до визначення ступеня подібності маршрутів та за допомогою метрики  $R(M_1, M_2)$

При використанні лінійної апроксимації координати  $(x, y)$  точки  $i$ -го сегменту прямої, який з'єднує точки  $(x_i, y_i)$  і  $(x_{i+1}, y_{i+1})$  першого маршруту, яка є найближчою до  $j$ -ї точки множини, координати якої рівні  $(x_j, y_j)$ , можна отримати з формул:

$$\begin{aligned} x &= \frac{y_j * k_j + x_j - b_i * k_i}{1 + k_i^2} \\ y &= k_i * x + b_i. \end{aligned} \quad (2.13)$$

де  $k_i, b_i$  – параметри лінійного рівняння прямої, що описує  $i$ -й сегмент, який з'єднує точки  $(x_i, y_i)$  і  $(x_{i+1}, y_{i+1})$  множини. Вказані параметри можуть бути нескладно знайдені з використанням рівняння прямої, яка проходить двома точками.

Тоді:

$$R_{minj} = \sqrt{\left(x_j - \frac{y_j * k_i + x_j - b_i * k_i}{1 + k_i^2}\right)^2 + \left(y_j - k_i * \frac{y_j * k_i + x_j - b_i * k_i}{1 + k_i^2} - b_i\right)^2}. \quad (2.14)$$

Формула може застосовуватись у випадку, коли найкоротша відстань від  $j$ -ї точки множини до встановленого першого маршруту знаходиться на перпендикулярі від  $j$ -ї точки множини до деякого сегмента неперервного апроксимованого першого маршруту.

Однак, можливий випадок, коли величина  $x$ , що отримана з (3), може знаходитись поза діапазоном. В такому випадку величину пропонується знаходити як найменшу відстань від  $j$ -ї точки множини до кінців відрізка  $i$ -го сегменту (тобто до точок  $(x_i, y_i)$  і  $(x_{i+1}, y_{i+1})$ ).

Тоді, з урахуванням цього, у загальному випадку

$$R_{minj} = \begin{cases} \sqrt{\left(x_j - \frac{y_j * k_i + x_j - b_i * k_i}{1 + k_i^2}\right)^2 + \left(y_j - k_i * \frac{y_j * k_i + x_j - b_i * k_i}{1 + k_i^2} - b_i\right)^2}, & x \in [x_i, x_{i+1}]. \\ \min \left\{ \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}, \sqrt{(x_j - x_{i+1})^2 + (y_j - y_{i+1})^2} \right\}, & x \notin [x_i, x_{i+1}]. \end{cases} \quad (2.15)$$

Послідовне використання виразів визначає методику обчислення  $R(M_1, M_2)$

Слід зауважити, що величина метрики  $R$ , яка використовується для оцінки ступеня подібності першого та другого маршрутів залежить від порядку слідування аргументів у записі для  $R$ . Так, якщо в записі  $R(M_1, M_2)$  аргументи та поміняти місцями, то запис метрики матиме вигляд  $R(M_2, M_1)$ , а її відшукування для того ж випадку початкових даних, що й наведені на рисунку нижче, може бути оцінене з рисунку.

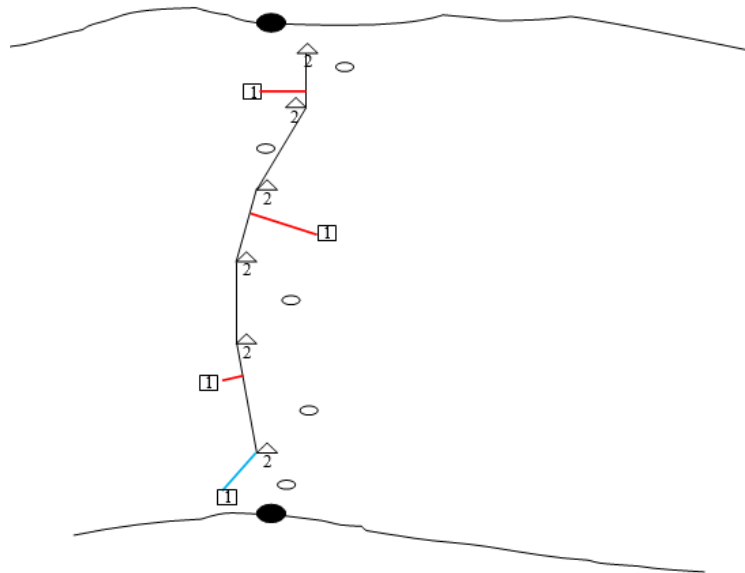


Рисунок 2.3 – Підхід до визначення ступеня подібності маршрутів та за допомогою метрики  $R(M_2, M_1)$

Для розрахунку  $R(M_2, M_1)$  необхідно насамперед здійснити встановлення (відновлення, генерування) неперервного маршруту руху другого судна між заданими у дискретній множині  $M_2$  точками з використанням лінійної апроксимації. Тоді величина  $R(M_2, M_1)$  визначається, як середнє відхилення точок множини  $M_1$ , які описують дискретний маршрут руху першого судна, від апроксимованого неперервного другого маршруту у вигляді

$$R(M_2, M_1) = \frac{\sum_{j=1}^{k_1} R_{\min j}}{k_1}. \quad (2.16)$$

де  $R_{\min j}$  є найкоротшою відстанню від  $j$ -ї точки множини  $M_1$  до встановленого другого маршруту, тобто до найближчого деякого  $i$ -го сегменту прямої, який з'єднує точки  $(x_i, y_i)$  і  $(x_{i+1}, y_{i+1})$  множини  $M_2$ , а  $k_1$  - кількість досліджуваних точок першого маршруту.

При використанні лінійної апроксимації сегментів другого маршруту координати  $(x, y)$  точки  $i$ -го сегменту прямої, який з'єднує точки  $(x_i, y_i)$  і  $(x_{i+1}, y_{i+1})$  другого маршруту, яка є найближчою до  $j$ -ї точки множини  $M_1$ , координати якої рівні  $(x_j, y_j)$ , також можна отримати за допомогою формул, але з урахуванням того, що  $k_i, b_i$  – параметри лінійного рівняння прямої, що описує  $i$ -й сегмент, який з'єднує точки  $(x_i, y_i)$  і  $(x_{i+1}, y_{i+1})$  множини  $M_2$ . Вказані параметри можуть бути нескладно знайдені з використанням рівняння прямої.

Тоді  $R_{\min j}$  можна визначати за допомогою формули у випадку, коли найкоротша відстань від  $j$ -ї точки множини  $M_1$  до встановленого другого маршруту знаходиться на перпендикулярі від  $j$ -ї точки множини  $M_1$  до деякого сегмента неперервного апроксимованого другого.

У випадку ж, коли величина  $x$ , знаходиться поза діапазоном  $[x_i, x_{i+1}]$ , величину пропонується знаходити як найменшу відстань від  $j$ -ї точки множини до кінців відрізка  $i$ -го сегменту (тобто до точок  $(x_i, y_i)$  і  $(x_{i+1}, y_{i+1})$  множини).

Аналіз підходів і даних, спонукає до необхідності перевірки рівності метрик  $R(M_1, M_2), R(M_2, M_1)$ . Адже, для однієї і тієї ж пари маршрутів у залежності від обраної метрики у разі неспівпадання  $R(M_1, M_2)$  і  $R(M_2, M_1)$  можуть мати місце різні висновки щодо подібності чи неподібності маршрутів. Саме тому для перевірки рівності величин  $R(M_1, M_2)$  і  $R(M_2, M_1)$  у роботі [1] було проведено окреме дослідження для вихідних даних.

Застосування формул для початкових даних дозволило отримати, що  $R(M_1, M_2) = 18$ ,  $R(M_2, M_1) = 26$ . А отже, у загальному випадку, дійсно  $R(M_1, M_2) \neq R(M_2, M_1)$ .

Відсутність властивості комутативності щодо метрик  $R^k$  і спонукала авторів роботи [1] до формування пропозицій щодо застосування нової метрики, яка б не містила недоліку, що стосувався б неоднозначності формування висновків відносно подібності маршрутів.

Ця пропозиція має вигляд метрики виду

$$R^k(M_1, M_2) = \min(R(M_1, M_2), R(M_2, M_1)). \quad (2.17)$$

Для перевірки властивості комутативності для метрики  $R^k$  було проведено дослідження для початкових даних, що наведені на рисунку. Результати дослідження, що описані у праці [3], вказують на комутативність метрики.

Однак питанню об'єктивності результатів, отриманих за допомогою метрики (2.16), у роботі [3] увага не приділялася.

Задача 2.

Введення деякого граничного порогового рівня  $r_{pk}$  щодо величини відхилення маршрутів і його порівняння з метрикою (2.16) дозволяє робити висновки про ступінь подібності маршрутів  $M_1$  і  $M_2$  незалежно від того, для якого з маршрутів формується апроксимаційна неперервна траєкторія руху, і відносно якого здійснюється оцінка близькості до такої траєкторії.

Встановлення геометричної близькості маршрутів за допомогою формули (2.16) дає можливість провести класифікацію та поділ на кластери з подальшим визначенням у кожному з них еталонного маршруту. Еталонні маршрути можуть використовуватись для оцінки ступеня

відхилення судна, відносно якого проводиться аналіз. Також може здійснюватись подальша класифікація нових даних, що поступають до системи, та оновлення або уточнення меж кластера.

Для проведення кластеризації маршрутів на основі наявних у базі «Гарт-12» даних у роботі [3] пропонується побудувати матрицю  $R$ , елементи якої визначаються наступним чином:

$$r_{ij} = \{ 0, i = j, R^k(M_i, M_j), i \neq j. \quad (2.18)$$

де  $M_i, M_j$  - маршрути відповідно  $i$ -го та  $j$ -го суден.

Після формування матриці  $R$  для кожного її рядка, тобто для  $i$ , обчислюється:

- кількість  $n_s$  елементів, які не перевищують порогового рівня  $r_{pk}$ , тобто кількість елементів, для яких у рядку виконується нерівність  $R^k(M_i, M_j) \leq r_{pk}$ ;
- величина  $s_v = \sum R^k(M_i, M_j)$ , яка формується лише з тих значень  $R^k(M_i, M_j)$ , для яких виконується умова  $R^k(M_i, M_j) \leq r_{pk}$ .

На основі отриманих даних, які сортуються за зменшенням  $n_s$  або за збільшенням формується впорядкована множина  $M_s$ , елементами якої є набори  $(i, n_s(i), s_v(i))$ .

Далі множина  $M_s$  використовується для визначення тих маршрутів, які можуть бути включені до одного і того ж кластера. Здійснюється це за допомогою наступного алгоритму:

- якщо для першого елемента множини  $M_s$  виконується умова  $n_s(i) > 0$ , то створюється кластер з центральним маршрутом  $i$  і в нього включаються всі маршрути  $j$ , для яких  $R^k(M_i, M_j) \leq r_{pk}$ .
- з множини  $M_s$  видаляються всі набори  $(i, n_s(i), s_v(i))$ , в яких перший елемент  $i$  співпадає із значеннями  $i$ , які визначили маршрути, що увійшли до створеного кластера.
- кроки 1-2 алгоритму повторюються, поки потужність множини  $M_s$  є більшою за 0.

Реалізація такого підходу дозволяє встановити всі кластери (коридори руху суден), які задовольняють умові не перевищення порогу  $r_{pk}$ .

Проведений вище аналіз задач 1, 2 дозволяє зробити висновки про те, що у них є багато моментів, які можуть впливати на достовірність результатів моделювання та зменшувати ефективність автоматизованої системи обробки даних СВНО.

До числа таких можна віднести:

1) припущення про доцільність застосування лінійної апроксимації при роботі з окремими сегментами між точками множини  $M_i = \{(x_1, y_1), \dots, (x_{k_i}, y_{k_i})\}$  при формуванні неперервного апроксимаційного маршруту чи  $M_1$ , чи  $M_2$ , є сумнівним і нічим не підтвердженим. Особливий сумнів виникає у випадку значної віддаленості в часі або просторі досліджуваних точок. А тому актуальності набуває завдання дослідження достовірності результатів на заданих експериментальних наборах даних у разі, якщо застосовуватиметься інший вид апроксимації (в тому числі за допомогою сплайнів). При цьому, окремої уваги потребує питання відповідності обраного методу апроксимації логічному змісту процесу руху суден;

2) дослідження маршрутів різних суден на предмет формування кластерів може здійснюватись не лише в межах між пунктами відправлення та призначення, а й між довільними двома точками, в межах яких існує особлива необхідність вивчення питання про порушення прикордонного законодавства. У цьому випадку застосування формули (2) для відшукування метрики  $R(M_1, M_2)$ , формули (6) для відшукування метрики  $R(M_2, M_1)$ , або формули (7) для встановлення метрики  $R^k(M_1, M_2)$ , потребує обґрунтування вибору кількостей  $k_2$  і  $k_1$  необхідних для дослідження точок, які б забезпечили достатній рівень достовірності висновків, що формуються;

3) для умов, що наведені вище, актуальним є питання і вибору для дослідження конкретних точок на маршрутах  $M_1$  та  $M_2$ , які при реалізації значень  $k_1$  і  $k_2$  забезпечували б достатній рівень достовірності висновків, що формуються. Актуальність дослідження цього питання обумовлюється тим, що при реалізації різних комбінацій досліджуваних точок величина  $R(M_1, M_2)$ , або  $R(M_2, M_1)$ , або  $R^k(M_1, M_2)$  може суттєво відрізнитися, що може впливати на достовірність методики кластеризації.

Викликає сумнів і питання забезпечення об'єктивності результатів, що визначаються застосуванням метрики (2.16). Адже властивість комутативності метрики справедлива не лише у випадку застосування (2.16). Так, при застосуванні метрики

$$R^k(M_1, M_2) = \max(R(M_1, M_2), R(M_2, M_1)), \quad (2.19)$$

або

$$R^k(M_1, M_2) = \frac{R(M_1, M_2) + R(M_2, M_1)}{2}, \quad (2.20)$$

або

$$R^k(M_1, M_2) = \sqrt{R(M_1, M_2)R(M_2, M_1)}. \quad (2.21)$$

існує велика ймовірність її комутативності, а отже, можливості застосування. Однак висновки щодо подібності маршрутів та співпадання меж кластерів при застосуванні наведених метрик у різних випадках можуть відрізнятися. А отже, питання адекватності застосовуваного інструментарію та достовірності результатів, що ним забезпечуються, потребують окремого вивчення.

При застосуванні метрик (2.11), або (2.15), або (2.16) може мати місце випадок, коли для окремих сегментів між точками  $(x_i, y_i)$  і  $(x_{i+1}, y_{i+1})$  множини або виконуватиметься умова  $x_i = x_{i+1}$ . Тобто у вибраній системі координат окремі сегменти можуть бути паралельними осі ординат. У цьому випадку формула (2.14) не може бути застосовною для відшукування величини  $R_{\min j}$ .

У разі, якщо деякий досліджуваний маршрут  $M_i$  на окремій ділянці маршруту між пунктом відправлення та призначення паралельний еталонному маршруту  $M_{bt}$ , але на цій ділянці величина  $R(M_i, M_{et}) > r_{pk}$ , то це може спричинити не включення маршруту  $M_i$  у кластер, хоча у загальному випадку цей маршрут може увійти до цього кластеру. Разом з тим, існуюча методика кластеризації цієї особливості не враховує. А оскільки така особливість важлива з

позиції формування висновків щодо наявності порушень прикордонного законодавства, то відповідне питання також потребує окремого дослідження.

У разі достатньої складності маршруту існуюча методика може бути незастосовною, оскільки може не відповідати логічному змісту досліджуваного процесу.

Формування еталонного маршруту Mbt описане у роботі [5]. Цей маршрут являє собою набір дискретних точок. Актуальним є питання формування такого методу побудови еталонного маршруту Mbt , який би забезпечував створення еталонного маршруту у вигляді неперервної траєкторії. Тому у даній роботі ми будемо покращувати даний метод кластеризації через нелінійну апроксимацію з використанням різних способів побудови маршруту судна.

### 2.3 Висновки

Проведене дослідження дозволяє зробити висновок, що основним положенням, яке реально визначає ефективність автоматизованої системи обробки даних, є метод кластеризації маршруту суден. Існуючий метод кластеризації характеризується низкою особливостей, які наведені вище та стосуються інструментальних особливостей реалізації та впливу на надійність результатів для подальшої оцінки ризиків незаконної діяльності на державному кордоні. Аналіз певних особливостей дозволяє зробити висновок, що існуючий метод кластеризації маршрутів кораблів не може явно бути придатним для використання в SVNO як процедурний модуль автоматизованої системи обробки даних, який забезпечив би достатній рівень надійності при виявленні ознак порушення прикордонного законодавства.

### 3 АЛГОРИТМИ ТА ТЕХНОЛОГІЯ РЕАЛІЗАЦІЇ УДОСКОНАЛЕНОГО МЕТОДУ КЛАСТЕРИЗАЦІЇ МАРШРУТІВ РУХУ СУДЕН

#### 3.1 Алгоритми реалізації удосконаленого методу кластеризації маршрутів руху суден

Алгоритми ієрархічної кластеризації. Серед ієрархічних алгоритмів кластеризації існує два основних типи: алгоритми визхідні та низхідні. Спадні алгоритми працюють «зверху-вниз»: на початку всі об'єкти розміщуються в одному кластері, який потім ділиться на менші та менші кластери. Більш поширені алгоритми знизу вгору, які на початку роботи розміщують кожен об'єкт в окремому кластері, потім відбувається об'єднання, поки всі об'єкти вибірки не опиняться в одному кластері. Так будується вкладена система розділення. Результати таких алгоритмів зазвичай подаються у вигляді дерева - дендрограми.

Щоб обрахувати відстані між кластерами часто використовують дві відстані: одинарне посилення або повне посилення.

Недоліком ієрархічних алгоритмів є повна система розділення, яка може бути зайвою в контексті вирішуваної проблеми.

Алгоритми квадратних помилок

Проблему кластеризації можна розглядати як побудову оптимального поділу об'єктів на групи. У цьому випадку оптимальність може бути визначена як вимога мінімізувати середньоквадратичну помилку розділу.

Проблему кластеризації можна розглядати як побудову оптимального поділу об'єктів на групи. У цьому випадку оптимальність може бути визначена як вимога мінімізувати середньоквадратичну помилку розділу:

$$e^2(X, L) = \sum_{j=1}^K \sum_{i=1}^{n_j} \|x_i^{(j)} - c_j\|^2. \quad (3.1)$$

де  $c_j$  - «центр мас» кластера  $j$ .

Алгоритми квадратичної помилки є одним із плоских алгоритмів. Найпоширенішим алгоритмом є метод k-середніх. Цей алгоритм будує задане число кластерів, максимально віддалених між собою.

Робота алгоритму:

- випадково вибрати k точок, які є «центрами мас» кластерів.
- віднести кожен об'єкт до кластеру з найближчим «центром мас».
- перерахувати «центри мас» кластерів відповідно до їх поточним складом.

Якщо критерій зупинки алгоритму не задоволений, повернутися до п. 2.

За критерій зупинки роботи алгоритму зазвичай вибирають мінімальне значення середньоквадратичної помилки. Так само можливо зупиняти роботу алгоритму, якщо на кроці 2 не було об'єктів, що перемістилися з кластера в кластер.

Найбільш популярним алгоритмом нечіткої кластеризації є алгоритм c-середніх (c-means). Він являє собою модифікацію методу k-середніх. Кроки роботи алгоритму:

- вибрати початкове нечітке розбиття n об'єктів на k кластерів шляхом вибору матриці приналежності U розміру n x k.
- використовуючи матрицю U, знайти значення критерію нечіткої помилки:

$$E^2(X, U) = \sum_{i=1}^N \sum_{k=1}^K U_{ik} \| x_i^{(k)} - c_k \|^2. \quad (3.2)$$

- де  $c_k$  - «центр мас» нечіткого кластера k:

$$c_k = \sum_{i=1}^N U_{ik} x_i. \quad (3.3)$$

- перегрупувати об'єкти з метою зменшення цього значення критерію нечіткої помилки.
- повертатися в п. 2 до тих пір, поки зміни матриці U не стануть незначними.

Усі вище представлені алгоритми по своєму допомагають реалізувати метод кластеризації. Проведений аналіз показує усі можливості даних алгоритмів, їхні переваги і недоліки. Також проаналізовано як з їхньою допомогою можна реалізувати метод кластеризації. Кожен з них по своєму виконує функцію реалізації методу. Кожний із алгоритмів може використовуватись тільки в тих умовах, де його переваги будуть якнайкраще і в повній мірі розкриті.

Тому далі будуть розглянуті та проаналізовані тільки алгоритми уже удосконаленого методу.

Основний алгоритм роботи удосконаленого методу кластеризації, а також основний алгоритм роботи програмного продукту має такий вигляд.

### Загальний алгоритм роботи



Рисунок 3.1-Загальний алгоритм роботи програми.

Як видно з алгоритму, спочатку відбувається введення основних базових даних про наявні маршрути руху суден. Після цього відбувається процес

підготовки і саме апроксимації усіх даних. Сама апроксимація являється нелінійною. В чому саме і полягає удосконалення методу. Наступним кроком є розрахунок даних про маршрути суден за відповідними формулами. Після проведення усіх потрібних розрахунків, програма виводить результати обрахувань.

Що ж до алгоритму самого методу, то він виглядає так:

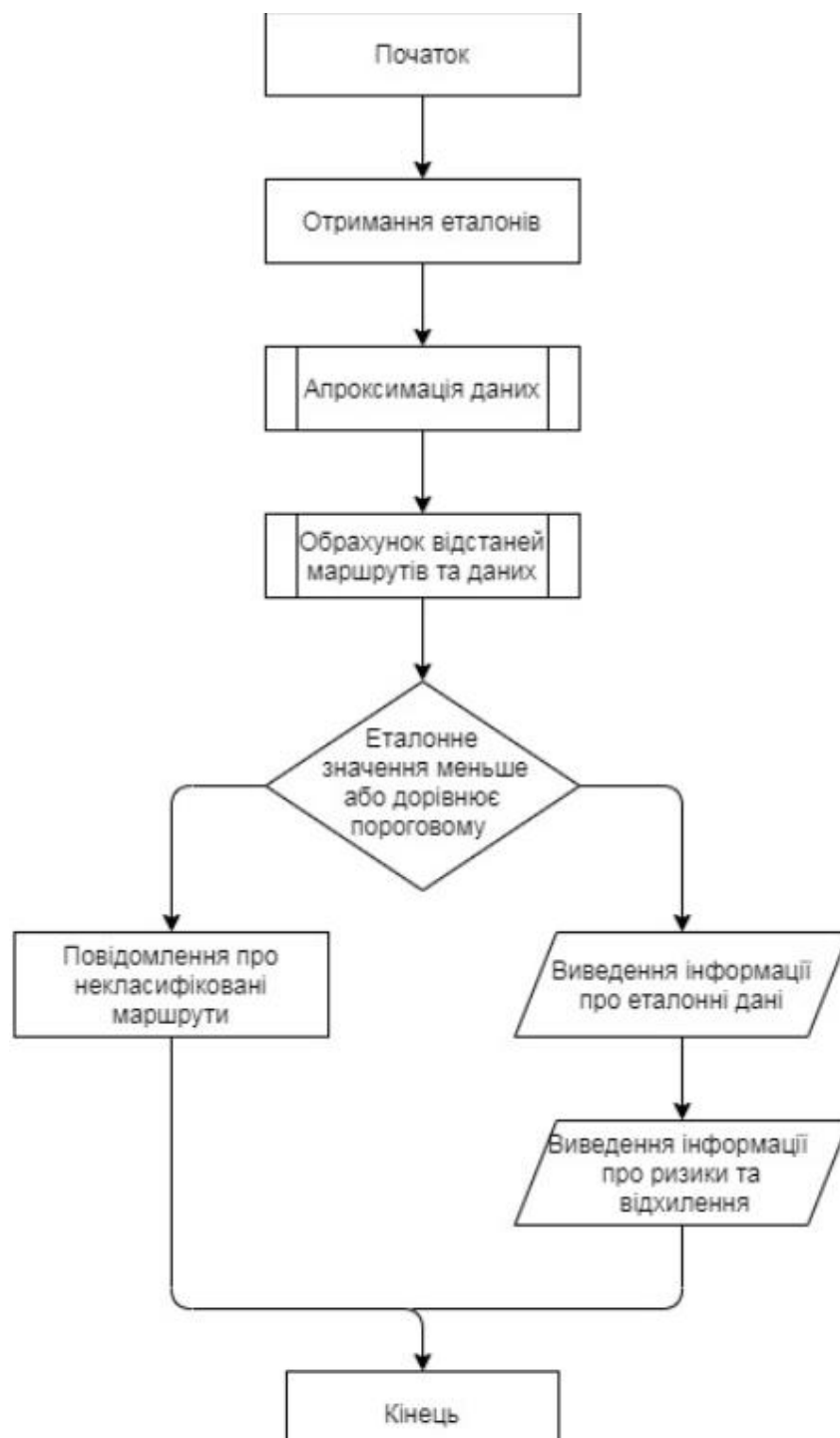


Рисунок 3.2 - Алгоритм побудови еталонного маршруту

Виходячи з алгоритму, процес побудови еталонного маршруту, для відслідковування руху судна відповідного до цього маршруту і його відповідний контроль, виконуються наступні кроки. Спочатку з усіх наявних маршрутів в кластері вибирається еталонний. Після цього відбувається апроксимація даних яка змінюється обрахунком відстаней маршрутів та даних. Виходячи з результатів цих обчислень, якщо еталонне значення менше, то виводиться повідомлення що маршрути не можуть бути класифіковані. Якщо ж еталонне значення дорівнює пороговому-виводиться інформація про еталонні дані. А також показується інформація про можливі ризики та відхилення від маршруту.

Усі ці алгоритми були програмно реалізовані і отримані результати а також аналіз самого коду буде у наступному розділі.

### 3.2 Проектування програмного забезпечення реалізації удосконаленого методу кластеризації маршрутів руху суден.

Отже, після того як ми ознайомились із алгоритмами реалізації побудови покращеного методу кластеризації, можна приступити до його програмної реалізації. Увесь процес написання програмного продукту буде відбуватись на мові програмування C#. Використовуючи усю відому нам інформацію, насамперед потрібно спроектувати саме програмне забезпечення.

Розглянемо основну програму де буде відбуватись побудова маршрутів.

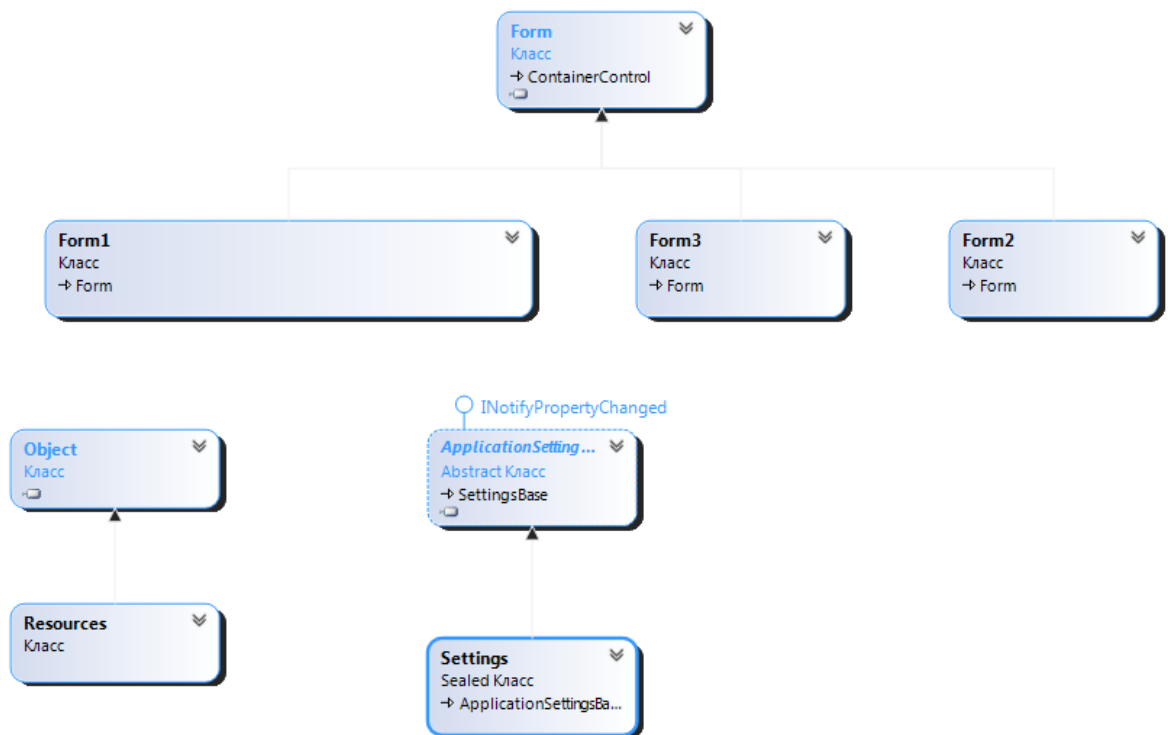


Рисунок 3.3-Діаграма класів системи висвітлення надводної обстановки

Виходячи з даної діаграми можна дізнатись вміст кожного наявного класу. Наприклад, розглянемо третю форму. Саме в ній відбувається введення даних по точки маршруту, з яких формується потім загальний маршрут.

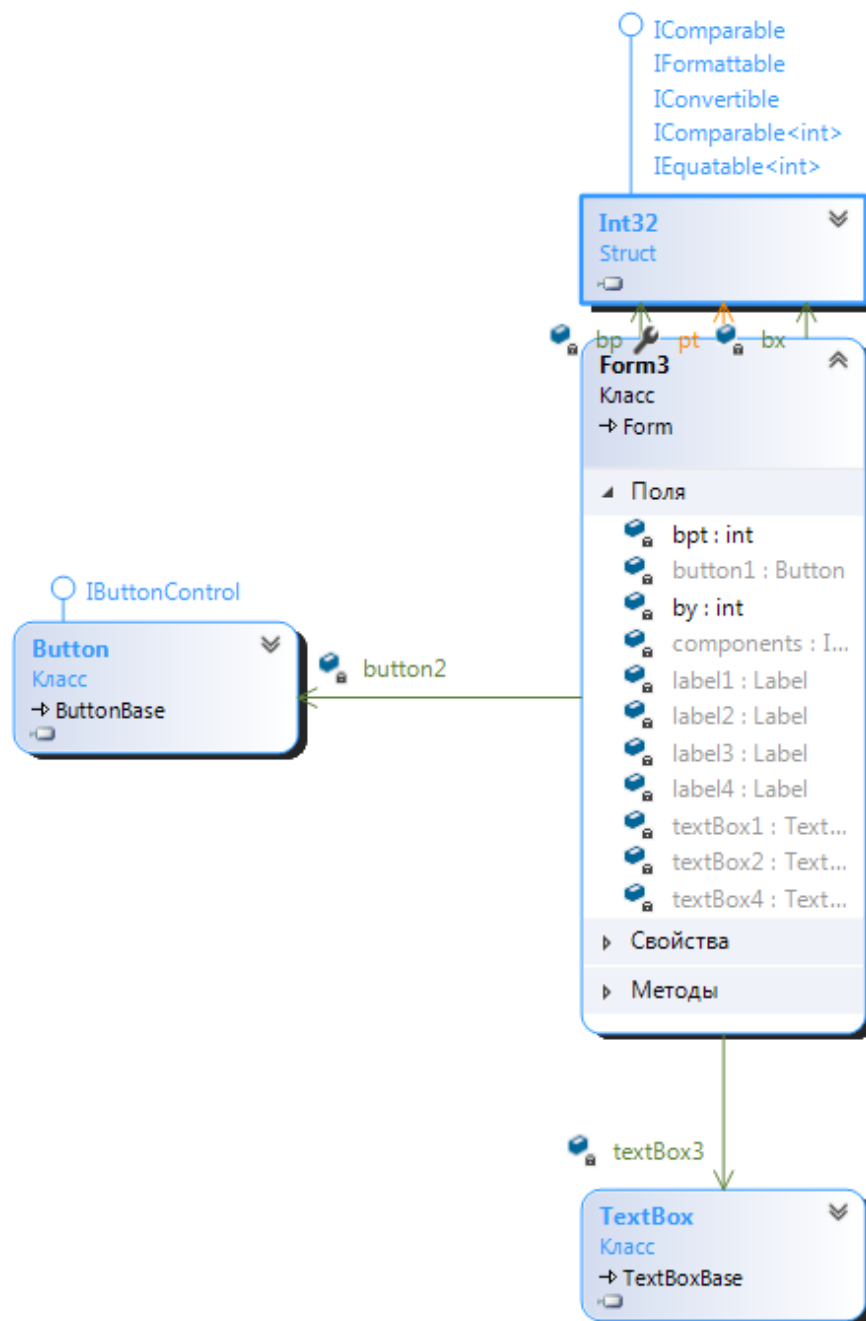


Рисунок 3.4-Діаграма асоціацій Form 3

Наведена нижче схема- це діаграма класів для реалізації для програмної реалізації побудови, на основі отриманих даних, еталонного маршруту.Алгоритм якого був показаний вище.

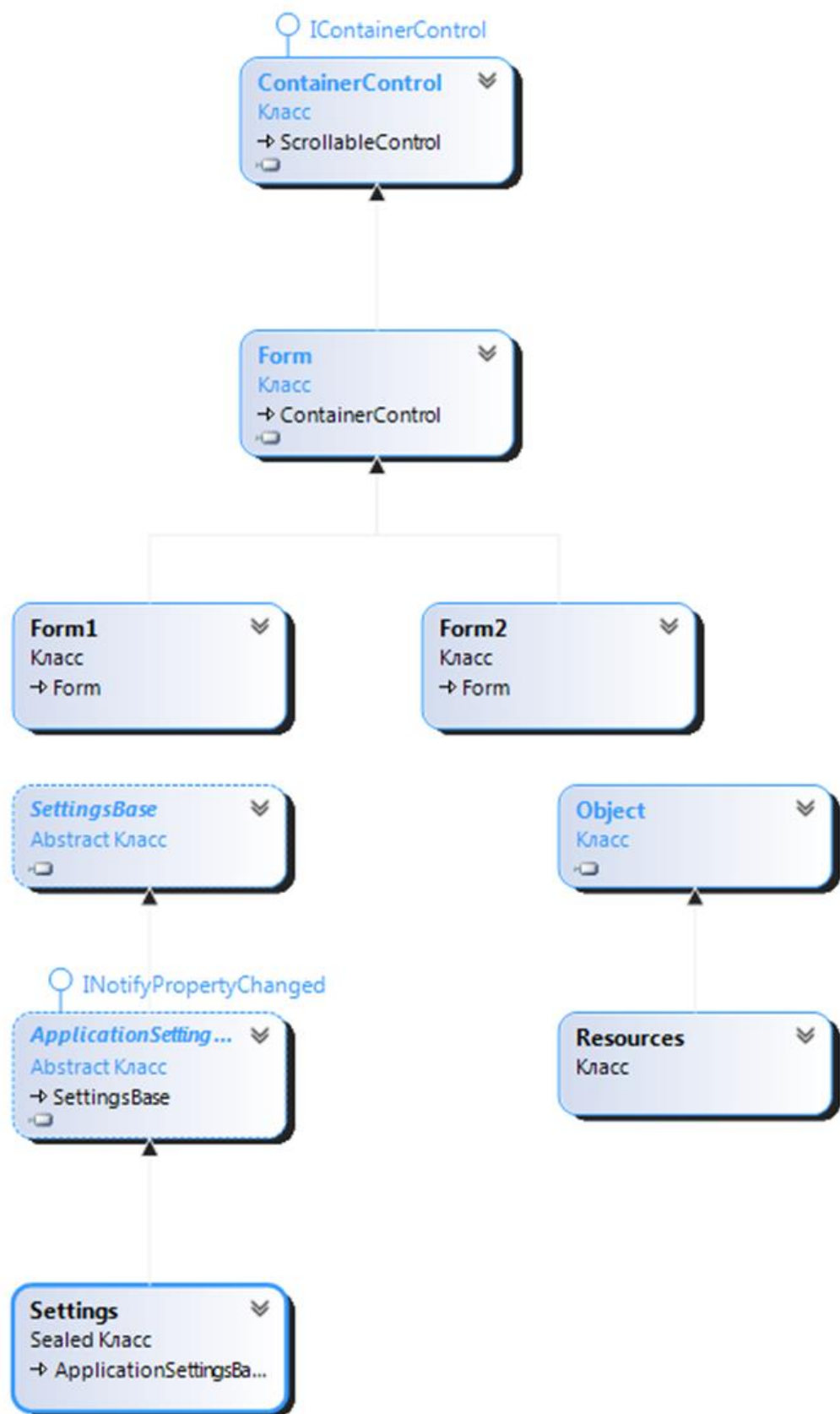


Рисунок 3.5-Діаграма класів

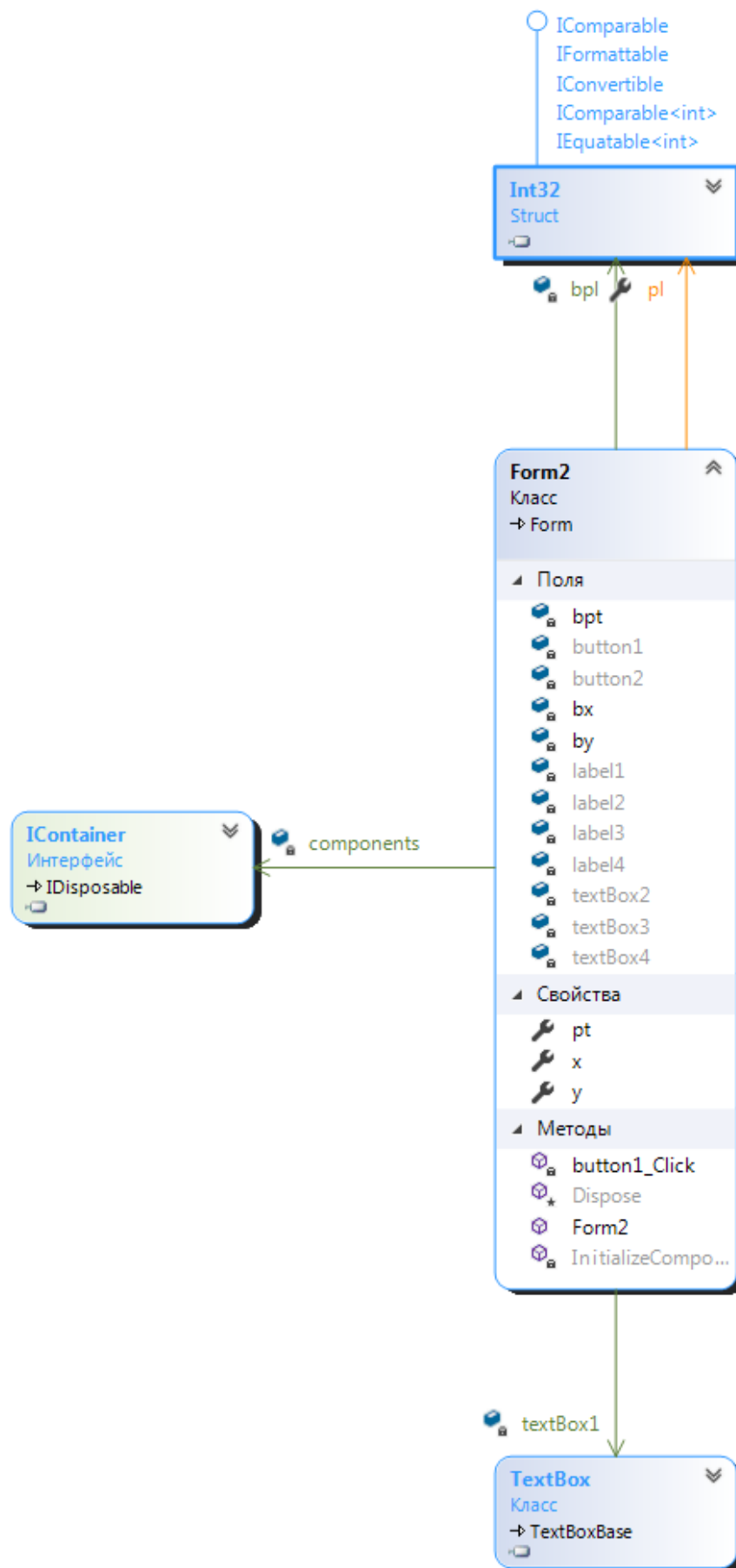


Рисунок 3.6-Діаграма асоціацій для Form1

### 3.3 Висновки

У даному розділі були розглянуті наявні алгоритми, за допомогою яких можна провести реалізацію методу кластеризації. Було проведено аналіз цих алгоритмів і розглянуто алгоритм, який реалізувався у створеній системі. Також було загально розглянуто кроки, які виконуються програмно для побудови еталонного маршруту по якому оцінюється рух корабля та виявляються просторові аномалії під час цього. Наведені діаграми для кращого розуміння архітектури програмного продукту. Уся ця інформація дає нам змогу зробити програмну реалізацію покращеного методу виявлення просторових аномалій на основі побудові еталонного маршруту через кластеризацію даних.

## **4 КІБЕРФІЗИЧНА СИСТЕМА ВИЯВЛЕННЯ ПРОСТОРОВИХ АНОМАЛІЙ РУХУ СУДЕН НА ОСНОВІ МЕТОДУ КЛАСТЕРИЗАЦІЇ ЇХ МАРШРУТІВ**

4.1 Опис середовища розробки кіберфізичної системи виявлення просторових аномалій руху суден

Сама кіберфізична система виявлення просторових аномалій руху суден розроблять в середовищі Visual Studio. Visual Studio це середовище для розробки програмних додатків від компанії Microsoft. Це середовище надає змогу розробляти будь-які програмні продукти, а саме:

- консольні програми;
- програми з графічним інтерфейсом;
- Web-додатки тощо.

Visual Studio дає змогу користувачам створити програму для роботи на декількох платформах Windows. А саме таких як мобільний, настільний і експериментальний середовища Microsoft HoloLen. Сюди також входить можливість написання додатків які не працюють на комп'ютер Windows, проте виконуються на пристроях iOS або у веб-додатках у хмарі.

Visual Studio включає в себе редактор коду, що підтримує компонент завершення коду і також рефакторинг коду. Цей відладчик працює на рівні вихідного, а також машинного рівнів. . Ще до вбудованих інструментів включають програму для кодування, конструктор форм для побудови графічних інтерфейсів, веб-дизайнер, дизайнер класів та дизайнер схеми баз даних. Він приймає плагіни, які покращують функціональність практично на всіх рівнях, включаючи підтримку систем керування джерельними ресурсами та додавання нових груп інструментів.

Саме середовище надає можливість працювати з різними мовами програмування. Це і Visual C#, Visual Basic, Visual F#, Visual C++, Python і т.д. Також існують інструменти для розробки програмних продуктів не тільки для платформи Windows, а і під інші популярні платформи: Android, iOS.

Дане середовище надає можливість підключити функціональність, кодовану як VSPackage. Після встановлення ця функціональність доступна у вигляді Сервісу. IDE надає три сервіси:

- SVsSolution, що забезпечує можливість переліку проектів та рішень;
- SVsUIShell, що забезпечує вікна та функціональність інтерфейсу користувача;
- SVsShell, що займається реєстрацією VSPackages.

Окрім того, IDE також відповідає за координування і забезпечення зв'язку між усіма службами. Visual Studio використовує COM для доступу до VSPackages.

Усі ці послуги можуть бути використанні для створення інших пакетів, які додають функціональності. Більш детально дане питання розкрив автор у роботі [65]. Також є підтримка мов програмування, яка додається за допомогою певного VSPackage, який називається службою мови. Мовна служба визначає різні інтерфейси, які реалізація VSPackage може здійснювати, щоб додати підтримку для різних функцій. Функціональні можливості, які можна так додати, включають синтаксичну окраску, завершення звітів, узгодження фігурних підказок, підказки про параметри інформації, списки учасників та маркери помилок для фонові компіляції. Якщо інтерфейс буде реалізовано, функціональність буде доступною для мови. Мовні служби реалізуються на кожній мові. Реалізації можуть повторно використовувати код з аналізатора або компілятора для мови. Дане питання розглядалось авторами роботи [66].

Створення програми. При створенні програм у Visual Studio користуються переважно двома підходами. Перший базується на основі створення окремих проектів для кожної задачі. В цьому випадку алгоритм роботи має такий вигляд:

1. Запускаємо середовище.
2. На початковій сторінці вибираємо Создать проект.
3. В верхній частині вікна вибираємо Visual C# --> Windows -->

Приложение Windows або Консольное приложение.

4. В нижній частині вікна вводимо ім'я проекту (поле Имя:); за допомогою кнопки Огляд вказуємо місце збереження проекту (поле Расположение:).

5. Натискаємо ОК.

Результатом цього алгоритму буде створення в папці MyProjects папки task\_1 із файлами проекту.

Розглянемо більш універсальний підхід. Нехай нам потрібно розв'язати три задачі (створити три додатки). Visual Studio дозволяє об'єднувати проекти в Рішення. Одне рішення може містити декілька проектів.

Алгоритм створення Рішення:

1. Запускаємо середовище розробки.  
2. На початковому екрані вибираємо Создать проект.  
3. В верхній частині вікна вибираємо Visual C# --> Windows --> Приложение Windows Forms або Консольное.

4. В нижній частині вікна вводимо ім'я першої задачі - першого проекту Task\_1 (поле Имя:);

5. Натискаємо ОК.

Ми створили рішення Lesson\_1 та проект Task\_1 в ньому. Далі, для додавання наступних проектів у рішення Lesson\_1 потрібно виконувати такі дії:

Пункт меню Файл --> Создать --> Проект... або [Ctrl]+[Shift]+[N] - з'явиться вже знайоме вікно Создание проекта.

Наступні дії аналогічні до попередніх.

Після того, як було створено новий проект Windows Forms, нам відразу видно вікно програми, яке називається форма. В майбутньому на ньому будуть розміщуватись різноманітні елементи програми. Незважаючи на те, що над проектом не було виконано ніяких дій, ми уже можемо його запускати. Після запуску нам буде показано просто пусте вікно з назвою Form1, після чого програму можна закрити. Якщо ж у вас є декілька проектів, пере запуском буде запропоновано вибрати необхідний проект ля відкриття.

Розглянемо деякі властивості Форми (табл. 4.1):

Таблиця 4.1- Властивості Форми у Visual Studio

Властивість	Пояснення
Text	Заголовок вікна
Size -Width -Height	Розмір форми
BackColor	Колір фону форми
BackgroundImage	Фоновий малюнок форми
BackgroundImageLayout	Спосіб відображення фонового малюнку форми: -Zoom -Stretch
StartPosition	Позиція форми на екрані після запуску: -Manual -CenterScreen

У випадку, якщо не потрібно встановлювати точні розміри об'єкта, його приблизні розміри можна задати просто мишею.

Відкриття рішення. Відкрити збережене колись рішення або проект можна, знайшовши і запустивши файл з розширенням .sln або виконавши у головному меню Visual studio команду: Файл --> Открыть --> Решение или проект... або [Ctrl]+[Shift]+[O] на клавіатурі.

Середовище розробки Visual Studio має багатовіконний інтерфейс який можна налаштовувати під потреби конкретного користувача.

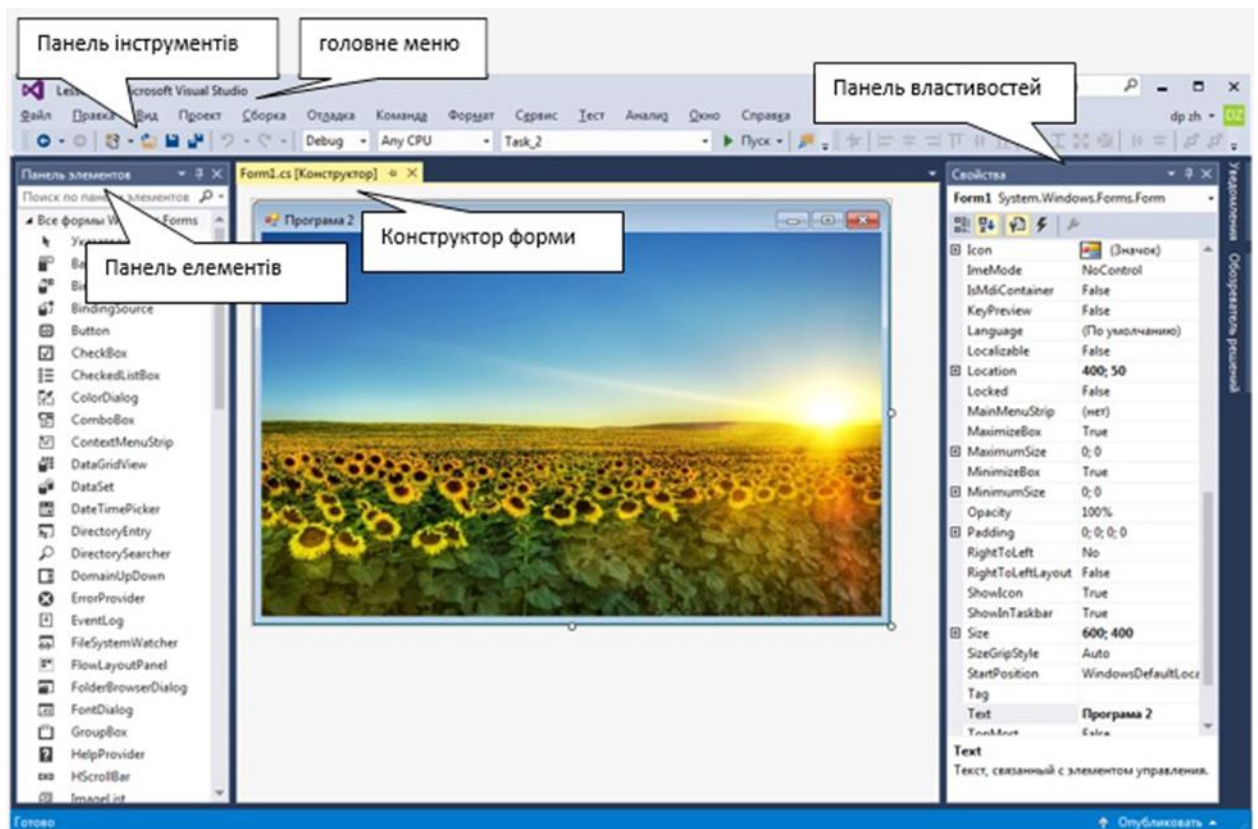


Рисунок 4.1-Інтерфейс Visual Studio.

Кількість та розташування панелей можна змінювати. Деякі панелі можуть бути згорнуті і відображатися в бокових панелях ліворуч або праворуч, наприклад, панель *Обозреватель решений*. Відкрити такі згорнуті панелі можна, натиснувши на їх назві. Проте, після виконання дій на такій панелі, вона автоматично буде згорнута. Для того, щоб закріпити панель на екрані, потрібно натиснути зображення «канцелярської кнопки».

Visual Studio не має такої можливості, як підтримка управління вихідним кодом, проте вона визначає два альтернативні шляхи для інтеграції систем керування вихідними кодами з IDE. Visual Studio підтримує запуск декількох екземплярів середовища (кожен зі своїм набором *VSPackages*) [67].

Visual Studio має додаткову компіляцію. У міру написання коду Visual Studio компілює його у фоновому 35 режимі, щоб надати відгук про помилки синтаксису та компіляції, позначені червоним хвилястим підкресленням. Попередження позначені зеленим підкресленням. Тональна компіляція не генерує виконуваний код, оскільки для цього потрібен інший компілятор, ніж той, який

використовується для створення виконуваного коду. Із цією функцією більше детально можна ознайомитись у роботі [68].

В загальному дане середовище розробки дає користувачеві різноманітні можливості і інструменти для реалізації його проєктів. При цьому також полегшується процес програмування, адже середовище містить велику кількість бібліотек, які допомагають у спрощені та покращенні програмного коду. Також наявна велика різноманітність програм на яких може кодити користувач. Однією з таких є C#. Саме за допомогою цієї мови програмування була реалізована систем для виявлення просторових аномалій руху суден.

Ця мова програмування була створена в кінці 1998 року компанією Microsoft. Її ціллю було надання можливості створення різноманітних програм для платформи Microsoft.NET. Така прив'язаність платформ Microsoft .NET та мови C# в подальшому була усунена, проте сама платформа .NET у всіх версіях містить компілятор кодів даної мови програмування. Це дозволяє запускати додатки без налаштування додаткового програмного забезпечення. Сама назва «C#» пов'язана з тим, що її синтаксис дуже нагадує C++. Загалом, C# перейняв багато позитивних рис своїх попередників - Delphi, C++, Java та інших. При цьому із C# були вилучені проблемні алгоритми. Мова C# актуальна в першу чергу тому, що дозволяє більш раціонально створювати популярні на сьогодні інтернет-додатки. C# тісно інтегрована з мовою XML, різноманітними веб-технологіями. Також популярність C# серед розробників обумовлена тим, що вона інтегрувала в собі переваги мов Java та C++, при чому з C# були виключені деякі сумнівні директиви, макроси, відмінені глобальні перемінні [67].

## 4.2 Програмна реалізація виявлення просторових аномалій руху суден

Загальний код усієї програми буде наведений у додатку. У цьому ж пункті буде розглянуто основні модулі, які відповідають саме за знаходження еталонного маршруту і методу кластеризації вцілому.

Основне вікно зображено на рисунку.

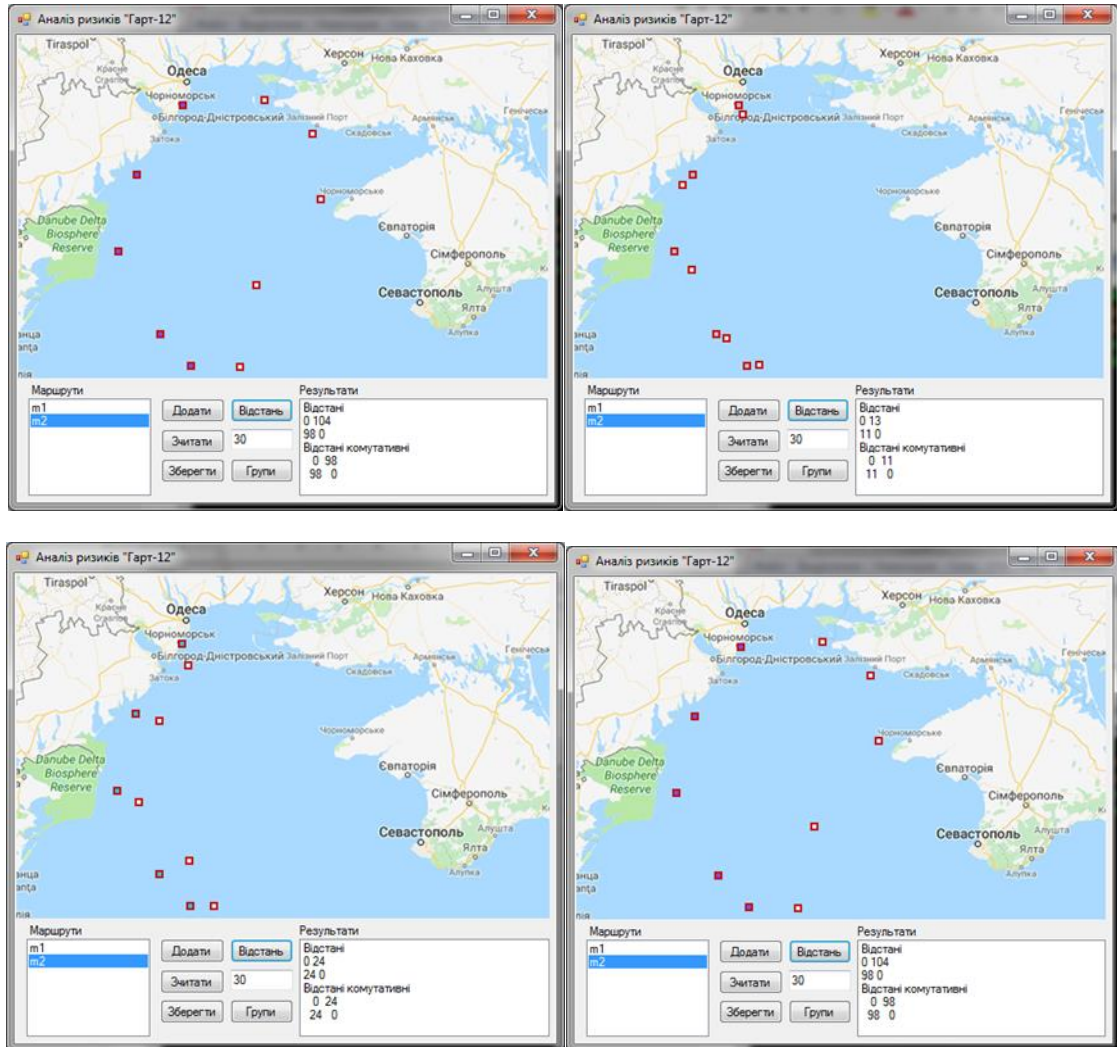


Рисунок 4.2 - Приклади основного вікна

Програмна реалізація функції обрахунку відстані:

```
for (int i = 0; i < nmar; i++)  
    for (int j = 0; j < nmar; j++)  
        if (i == j) metr[i, j] = 0;
```

```

else
{
    tmar m1=(tmar)listBox1.Items[i];
    tmar m2=(tmar)listBox1.Items[j];
    float r = 0;
    if (m2.pbcs.Count>0)
    for (int l = 0; l < m2.pbcs.Count; l++)
    {
        int x1 = m2.pbcs[l].x;
        int y1 = m2.pbcs[l].y;
        float min = 300000;
        for (int m = 0; m < m1.pbcs.Count-1; m++)
        {
            float k=m1.pbcs[m].k;
            float b=m1.pbcs[m].b;
            float r1=0;
            int mx = (int)((k * y1 + x1 - k * b) / (1 + k * k));
            int my = (int) (k * mx+b);
            if (((mx > m1.pbcs[m].x) & (mx < m1.pbcs[m + 1].x)) |
((mx > m1.pbcs[m+1].x) & (mx < m1.pbcs[m ].x)))
            {
                r1 = (float)Math.Sqrt(Math.Pow(mx - x1, 2) +
Math.Pow(my - y1, 2));
            }
            else
            {
                float r2 = (float)Math.Sqrt(Math.Pow(m1.pbcs[m].x - x1,
2) + Math.Pow(m1.pbcs[m].y - y1, 2));
                float r3 = (float)Math.Sqrt(Math.Pow(m1.pbcs[m+1].x -
x1, 2) + Math.Pow(m1.pbcs[m+1].y - y1, 2));
            }
        }
    }
}

```

```
        if (r2 < r3) r1 = r2; else r1 = r3;
    }
    if (r1 < min) min = r1;
}
r += min;
}
if (m2.pbc.Count > 0) r = r / m2.pbc.Count; else r = 30000;
metr[i, j] = (int)r;
}
```

У наведеному вище уривку програмного коду, показано як саме відбувається обрахунок відстаней у системі виявлення просторових аномалій руху суден.

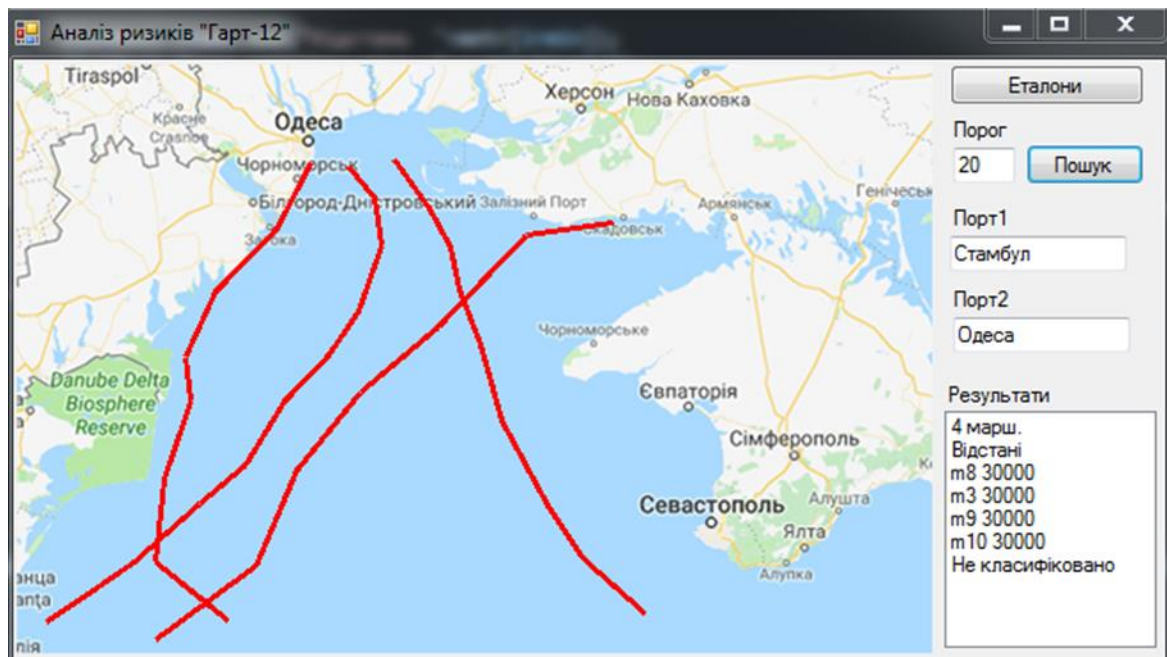


Рисунок 4.3-Основне вікно

Як саме відбувається апроксимація можна розглянути на прикладі уривку з коду.

```
foreach (tmar mar in etalon)
{
if (mar.pbcs.Count > 0)
for (int i = 0; i < mar.pbcs.Count - 1; i++)
{
mar.pbcs[i].k = ((float)(mar.pbcs[i + 1].y - mar.pbcs[i].y)) / ((float)(mar.pbcs[i +
1].x - mar.pbcs[i].x));
mar.pbcs[i].b = mar.pbcs[i].y - mar.pbcs[i].k * mar.pbcs[i].x;
```

Вивід результатів:

```
listBox1.Items.Add("Відстані");
for (int j = 0; j < nmar; j++) listBox1.Items.Add(etalon[j].name+" "+ metr[
j].ToString());
```

```

int por = int.Parse(textBox1.Text);
int rmin = metr[0];
int irmin = 0;
for (int i=1;i<nmar;i++)
if (metr[i] < rmin)
{
rmin = metr[i];
irmin = i;
}
if (rmin <= por)
{
listBox1.Items.Add("Класифіковано");
listBox1.Items.Add(etalon[irmin].name);

bool oz1;
if (((strdist(textBox2.Text, etalon[irmin].port1) < 15) & (strdist(textBox3.Text,
etalon[irmin].port2) < 15))|
((strdist(textBox2.Text, etalon[irmin].port2) < 15) & (strdist(textBox3.Text,
etalon[irmin].port1) < 15))) oz1=true; else oz1=false;
if (oz1) listBox1.Items.Add("Опису відповідає"); else
listBox1.Items.Add("Опису невідповідає");

```

### 4.3 Висновки

У даному розділі було представлено алгоритми реалізації знаходження еталонного маршруту, а також загальний алгоритм роботи системи. Було розглянуто і проаналізовано загальні відомості про середовище розробки в якому відбувався увесь процес. Відбулось ознайомлення із видом мови програмування на якій проводилась уся робота. Також був показаний основний код, за яким програма формує еталонний маршрут. Виходячи з доступної інформації, програма надає змогу виявленню просторових аномалій під час руху судна. До цього входить повідомлення про можливі ризики і відслідковування маршруту корабля.

## ВИСНОВКИ

У першому розділі було проведено аналіз наявних методів і засобів виявлення просторових аномалій руху суден. Були виділені їхні основні переваги і недоліки. Також була поставлена задача усїєї даної роботи. Проаналізувавши всю доступну інформацію, було запропоновано розглянути метод кластеризації даних для його подальшого використання у таких системах.

У другому розділі були розглянуті наявні підходи до виявлення просторових і часових аномалій. Також був більш детально розглянутий і описаний метод кластеризації даних. Був запропонований спосіб покращення наявного методу, на основі якого на даний момент відбувається функціонування системи оцінки надводної обстановки «Гарт-12».

У третьому розділі були розглянуті наявні алгоритми реалізації методу кластеризації. Також було алгоритмічно показано як саме буде відбуватися процес реалізації удосконаленого методу. Також був показаний алгоритм загальної роботи системи для висвітлення надводної обстановки і інформації про маршрути і рух корабля.

У четвертому розділі було розглянуто середовище у якому відбувався увесь процес реалізації системи. Також відбулося ознайомлення і мовою на якій відбувалося програмування. Була показана основна логіка в коді, а саме удосконалений метод виявлення просторових аномалій, шляхом формування еталонного маршруту.

Отже задача яка була поставлена на самому початку роботи, а саме покращити метод і програмно його реалізувати-виконана.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Боровик О. В., Рачок Р. В., Мазур В. Ю. Визначення підходів до просторового аналізу даних в інформаційно-телекомунікаційній системі морської охорони «Гарт-12». *Військові та технічні науки*. Хмельницький, 2018. № 1. С. 134-143.
2. Мазур В. Ю., Боровик О. В. Методичні основи оцінки ефективності функціонування єдиної системи висвітлення надводної обстановки на морській ділянці. *Наука і техніка Повітряних Сил Збройних Сил України*, 2018. № 2 . С. 182-189.
3. Мазур В. Ю., Боровик О. В., Рачок Р. В. Метод кластеризації маршрутів суден в системі висвітлення надводної обстановки . *Сучасні інформаційні технології у сфері безпеки та оборони*. 2018. № 2 – С. 87-92.
4. Pan Sheng, Jingbo Yin Extracting Shipping Route Patterns by Trajectory Clustering Model Based on Automatic Identification System Data. *Sustainability*, 2018, С. 10.
5. Боровик О. В., Рачок Р. В., Мазур В. Ю. Методика виявлення просторових аномалій руху суден та її використання при оцінюванні ризиків у системі морської охорони «Гарт-12». *Наука і оборона*, 2018. № 2. С. 65-69.
6. Arguedas V. F., Pallotta G., Vespe M. Maritime Traffic Networks. Historical Positioning Data to Unsupervised Maritime Traffic Monitoring. *Transactions on intelligent transportation systems*, , № 3, 2018, P. 722-732.
7. О. М. Ткаченко, О. Ф. Грійо Тукал, О. В. Дзіс, С. М. Лаховець. Метод кластеризації на основі послідовного запуску k-середніх з удосконаленням вибором кандидата на нову позицію вставки. 112-157.
8. Benjamin S. Duran, Patrick L., Odell. Cluster analysis a survey. P.654-705.
9. Vilskiy G. Cluster-probabilistic methodology for the study of maritime traffic information security. *Ukrainian Scientific Journal of Information Security*, 2014, vol. 20, issue 1, p. 92-96.

10. Кондратенко, Ю. П.: Нечіткі моделі та алгоритми для прогнозування параметрів морських контейнерних перевезень. 2008.С. 85-123.
11. 1Asa Ben-Hur., David Horn., Hava T. Siegelmann., Vladimir Vapnik. Support Vector Clustering. *Journal of Machine Learning Research* 2 2001, 125-137.
12. Падакін, Д. Ю.: Оптимізація систем берегового навігаційного обладнання. 2009. С. 54-82.
13. Казак Ю. В.: Розробка способу урахування траєкторної похибки повороту судна при оцінці безпеки судноводіння. С. 8-52.
14. Lior Rokach., Oded Maimon. Clustering methods. P. 26-99.
15. Геннадій В. Дослідження інформаційного забезпечення систем управління рухом морських суден. *Ukrainian Scientific Journal of Information Security*, 2013, P. 68-94.
16. Тихонов І.В.. Методологічні основи поліергатичного забезпечення авіації та управління рухом водних транспортних засобів(цільова техніка безпеки). Національний авіаційний університет, 2018, С. 19-32.
17. XU, Dongkuan; TIAN, Yingjie. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2015, 165-193.
18. Чайковський, І. В. Основні принципи побудови системи математичного забезпечення задач контролю і аналізу роботи флоту. *Вісник Східноукраїнського національного університету імені Володимира Даля*, 2019, 212-215.
19. Ворохобин, и. И. Impact of the cross-track error distribution law on safe navigation in narrow waters .С. 35-86.
20. Einstein, A., B. Podolsky, and N. Rosen, 1935, "Can quantum-mechanical description of physical reality be considered complete?", 777-780.
21. Рачок, Р. Формування концепції розбудови геоінформаційної системи у сфері забезпечення прикордонної безпеки. Збірник наукових праць Національної академії Державної прикордонної служби України. Сер.: Військові та технічні науки, 2017, 295-310.

22. Бень, А. П.; Паламарчук, І. В. Особливості побудови сучасних високоточних інтелектуальних систем управління рухом морських суден. *Науковий вісник Херсонської державної морської академії*, 2016, 4-10.
23. Вагущенко Л. Л. Современные информационные технологии в судовождении . ОНМА, С. 2013. – 135.
24. Вагущенко Л. Л. Расхождение с судами смещением на параллельную линию пути. Монография . Фенікс , 2013,С. – 180.
25. Вагущенко Л. Л. Алгоритм выработки рекомендаций по расхождению с судами . Судовождение : сб. науч. трудов ОНМА. – Одесса , 2012. С. 42–50.
26. Алексишин А. В. Расчет граничных пеленгов к зонам безопасности судов . *Науковий вісник Херсонської державної морської академії* .2012, С. 8–15.
27. Алексишин А. В. Выявление ситуаций опасного сближения судов с помощью граничных пеленгов в автоматизированных комплексах судовождения .*Науково- виробничий журнал ОНМУ*.2013, С. 117–123.
28. 8. Bi X.Y., Liu X.J. (2015) Research on Double Collision Avoidance Mechanism of Ships at Sea. *TransNav. the International Journal on Marine Navigation and Safety of Sea Transportation*, No. 1, pp. 13-16.
29. Lisowski J. (2014) Game Strategies of Ship in the Collision Situations. *TransNav. the International Journal on Marine Navigation and Safety of Sea Transportation*, No. 1,pp. 69-77.
30. Падакін, Д. Ю.; Падакин, Д. Ю. Оптимізація систем берегового навігаційного обладнання. 2009. С.26-52.
31. Єлезаров, О. П.; Масік, І. П. Синтез методу оцінки ступеня небезпеки ситуацій під час руху судна. Системи обробки інформації, 2019, С. 140-145.
32. Шаблій, Гліб. Система спостереження за надводною обстановкою та засобами навігаційного обладнання–крок до підвищення безпеки мореплавства. *Вісник держгідрографії*, 2014,. С.25-39.
33. Gladkih, Igor; Karochkina, Margaryta. Фізико-статистичне моделювання гідродинамічного поля корабля на базі гідродинамічних

розрахунків. *Вісник Національного технічного університету «ХПІ»*. 2018, С. 98-105.

34. Valdir A. Pereira; Matheus F. Sanches; Jordan F. Saran; Caio S. Coneglian; Leonardo C. Botega; Regina B. de Araujo. Towards semantic fusion using information quality and the assessment of objects and situations to improve emergency situation awareness. Eleventh International Conference on Digital Information Management (ICDIM).2016, C.59-84.

35. Zhiheng Wang; Zhuowei Hu; Hongqi Liu; Huili Gong; Wenji Zhao; Mengliang Yu; Mingzhi Zhang. Application of the relief degree of land surface in landslide disasters susceptibility assessment in China. International Conference on Geoinformatics 2010. P.26-88.

36. Lei Cheng-hua; Liu Gang; Liu Yi-gang. An accuracy assessment method of calculating cable conductor temperature through surface temperature and actual loading current. IEEE International Symposium on Electrical Insulation.2010 P.

37. Miao Qun; Gao Ying; Liu Zhiqiang; Tan Xiaohui Application of Comprehensive Water Quality Identification Index in Water Quality Assessment of River. WRI Global Congress on Intelligent Systems.2009. P.26-79.

38. Ji-hong Zhou; Cai-lian Hao; Jun-guang Zhao; Ping Li. Water Quality Assessment of Xida Section Based on Fuzzy Comprehensive Evaluation. Third International Symposium on Intelligent Information Technology Application Workshops.2009. P.20-45.

39. Yi-Chao Cai; Ling He; Wei-Ming Zhang; Zhong Liu. A Situation and Threat Assessment Model Based on Group Analysis. International Conference on Machine Learning and Cybernetics.2005.P. 29-35.

40. Katsunori Oyama; Kazuki Nakamura; Hiroyuki Wakabayashi; Atsushi Takeuchi; Naoki Ishitsuka. Situational map integration of dose distribution on the ground surface using Unmanned Aerial Vehicles. IEEE International Multi-Disciplinary Conference on Cognitive Methods in [::Situation::] Awareness and Decision Support (CogSIMA)2013. P.67-125.

41. Min-Gi Jeong; Eun-Bang Lee; Moonjin Lee. An Adaptive Route Plan Technique with Risk Contour for Autonomous Navigation of Surface Vehicles. OCEANS 2018 MTS/IEEE Charleston. P.97-246.
42. Yu-Hu Zhang; Hai-feng Jia; Cong Yang. Assessment of spatial-temporal variations in surface water quality of Luzhi river system in Plain River-Net Areas, Suzhou, China.2011. P.86-162.
43. Changjun Zhu; Xiujuan Zhao; Jihong Zhou. ANN based on PSO for surface water quality evaluation model and its application. Chinese Control and Decision Conference.2009. P.56-123.
44. David C. Burnett; Donald Sheaffer. Preliminary evaluation of imaging sonars for high-security underwater assessment. OCEANS 2010 MTS/IEEE SEATTLE. P.86-178.
45. L. Madureira; M. Pinho; P. L. Costa; S. Weigert; G. Griep; L. Calliari. Acoustical database dedicated to fish stock assessment being optimized to generate seafloor information and integrate biological, environmental and geomorphological studies. IEEE/OES Acoustics in Underwater Geosciences Symposium.2013. P.36-96.
46. Eric Munday; Tim Acker; James Dawson. Specialized tools for biological assessment using split beam hydroacoustics. OCEANS - San Diego.2013. P.56-125.
47. Niroj Sapkota; Abeer Alsadoon; P.W.C. Prasad; Amr Elchouemi; Ashutosh Kumar Singh. Data Summarization Using Clustering and Classification: Spectral Clustering Combined with k-Means Using NFPH. International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon).2019. P.23-94.
48. Hung-Leng Chen; Kun-Ta Chuang; Ming-Syan Chen. On Data Labeling for Clustering Categorical Data. IEEE Transactions on Knowledge and Data Engineering.2008. P.12-92.
49. Nisha; Puneet Jai Kaur. Cluster quality based performance evaluation of hierarchical clustering method. 1st International Conference on Next Generation Computing Technologies (NGCT).2015. P.51-88.

50. Amir Khoshkbarchi; Ali Kamali; Mehdi Amjadi; Maryam Amir Haeri. A modified hybrid Fuzzy clustering method for big data. 8th International Symposium on Telecommunications (IST).2016. P.65-91.
51. Ni Bin. Research on Methods and Techniques for IoT Big Data Cluster Analysis. International Conference on Information Systems and Computer Aided Education (ICISCAE).2018. P.94-126.
52. Xiaolu Zhu; Jinglin Li; Zhihan Liu; Fangchun Yang. A Joint Grid Segmentation Based Affinity Propagation Clustering Method for Big Data. IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS).2018. P. 23-46.
53. Sen Wu; Shujuan Gu. A Cluster Description Method for High Dimensional Data Clustering with Categorical Variables. International Conference on Measuring Technology and Mechatronics Automation.2010. P. 25-86.
54. Jin Zhou; Yuqi Pan; C. L. Philip Chen; Dong Wang; Shiyuan Han. K-medoids method based on divergence for uncertain data clustering. IEEE International Conference on Systems, Man, and Cybernetics (SMC).2016. P. 78-121.
55. Cheng-Fa Tsai; Sheng-Chiang Huang. An effective and efficient grid-based data clustering algorithm using intuitive neighbor relationship for data mining. International Conference on Machine Learning and Cybernetics (ICMLC).2015. P.85-215.
56. Chintan Shah; Anjali Jivani. Comparison of data mining clustering algorithms. Nirma University International Conference on Engineering (NUiCONE).2013. P.14-27.
57. Xiao-ping Lin; Zheng-yuan Mao; Jian-hua Liu. A Spatial Clustering Method by Means of Field Model to Organize Data. Second WRI Global Congress on Intelligent Systems.2010. P.86-167.
58. Hamza Erol; Bala Mikat Tyoden; Recep Erol. Classification Performances Of Data Mining Clustering Algorithms For Remotely Sensed Multispectral Image Data. Innovations in Intelligent Systems and Applications (INISTA).2018. P.67-278.

59. Троелсен Э. Язык программирования C# и платформа .NET 2014, С. 208.
60. Нейгл К. C# 5,0 и платформа .NET 4.5 для профессионалов. 2013, С. 1440.
61. Просиз Д. Программирование для Microsoft .NET . 2003, С. 704.
62. Скит Д. C# для профессионалов: тонкости программирования: 3-е издание, новый перевод C# in . 2014, С. 608.
63. Хейлсберг А., Торгерсен М., Вилтамут С., Голд П. Язык программирования для Microsoft .NET. Классика Computers Science, .2012. С. 784.
64. Стилмен Э. Изучаем C# . 2003, С. 704.
65. Албахари Д. C# 5.0. Справочник. Полное описание языка. 2013. С. 1008.
66. Уотсон К., Нейгел К., Язык программирования C#: полный курс. — М.: Диалектика, 2010. С. 288.
67. Mickey Williams Microsoft Visual C# (Core Reference) – Microsoft Press Redmond, WA, USA, 2002. P. 750.
68. Jeff Proise Programming Microsoft .NET – Microsoft Press Redmond, WA, USA, 2002. P. 800.
69. Wikipedia. Інсталяція ПЗ. URL: [https://uk.wikipedia.org/wiki/Інсталяція\\_ПЗ](https://uk.wikipedia.org/wiki/Інсталяція_ПЗ).
70. Боровик О.В., Боровик Д.О., Костельна Т. В. Щодо необхідності удосконалення методу кластеризації маршрутів суден, як процедурного модуля автоматизованої системи обробки даних в системі висвітлення надводної обстановки. Computer Systems and Information Technologies. 2020. №2. С. 38-46

## ДОДАТОК А

(обов'язковий)

### ЛІСТИНГ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВИЯВЛЕННЯ ПРОСТОРОВИХ АНОМАЛІЙ РУХУ СУДНА

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Xml;

namespace Maz1
{

    public partial class Form1 : Form
    {

        public class pb : PictureBox
        {
            public int x, y;
            public int lp, tp;
            public float k, b;
        }

        public class tmar
        {
```

```
public string name;
public string port1;
public string port2;
public List<pb> pbcs;
public override string ToString()
{
    return name;
}
}
```

```
private int mx, my, pbcnt;
private bool canMove;
private Point ppp;
```

```
public Form1()
{
    InitializeComponent();
    //listBox1.Items.Add(new tmar());
}
```

```
private void button1_Click(object sender, EventArgs e)
{
    Form2 fm = new Form2();
    if (fm.ShowDialog(this) == DialogResult.OK)
    {
        tmar tm = new tmar { name = fm.name, port1=fm.port1, port2=fm.port2 };
        tm.pbcs = new List<pb>();
        listBox1.Items.Add(tm);
    }
}
```

```
}
```

```
private void pictureBox1_MouseDoubleClick(object sender, MouseEventArgs e)
```

```
{
```

```
    tmar mr;
```

```
    if (listBox1.SelectedIndex >= 0)
```

```
    {
```

```
        mr = (tmar)listBox1.SelectedItem;
```

```
        mx = e.X * pictureBox1.Image.Width / pictureBox1.Width;
```

```
        my = e.Y * pictureBox1.Image.Height / pictureBox1.Height;
```

```
        pb pb2;
```

```
        pb2 = new pb();
```

```
        pb2.x = e.X + pictureBox1.Left - 4;
```

```
        pb2.y = e.Y + pictureBox1.Top - 4;
```

```
        pb2.Location = new System.Drawing.Point(e.X + pictureBox1.Left - 4, e.Y +  
pictureBox1.Top - 4);
```

```
        pbcnt++;
```

```
        pb2.Name = "pb" + pbcnt.ToString();
```

```
        pb2.Size = new System.Drawing.Size(8, 8);
```

```
        pb2.SizeMode
```

```
System.Windows.Forms.PictureBoxSizeMode.StretchImage;
```

```
        pb2.TabIndex = 1;
```

```
        pb2.TabStop = false;
```

```
        this.Controls.Add(pb2);
```

```
        pb2.BringToFront();
```

```

        pb2.MouseMove                +=                new
System.Windows.Forms.MouseEventHandler(this.pb1_MouseMove);
        pb2.MouseDown                +=                new
System.Windows.Forms.MouseEventHandler(this.pb1_MouseDown);
        pb2.MouseUp                  +=                new
System.Windows.Forms.MouseEventHandler(this.pb1_MouseUp);
        pb2.MouseDoubleClick          +=                new
System.Windows.Forms.MouseEventHandler(this.pb1_MouseDb1);

```

```

        Bitmap Bmp = new Bitmap(8, 8);
        using (Graphics gfx = Graphics.FromImage(Bmp))
        {
            using (SolidBrush brush = new
SolidBrush(System.Drawing.Color.FromArgb(192, 16, 16)))
            {
                gfx.FillRectangle(brush, 0, 0, 8, 8);
            }
            using (SolidBrush brush = new
SolidBrush(System.Drawing.Color.FromArgb(255, 228, 235)))
            {
                gfx.FillRectangle(brush, 2, 2, 4, 4);
            }
        }
        pb2.Image = Bmp;
// listBox2.Items.Add(pb2.x.ToString() + " " + pb2.y.ToString());
        mr.pbcs.Add(pb2);
// pbcs.Add(pb2);
    }
}

```

```

private void pb1_MouseDown(object sender, MouseEventArgs e)
{
    canMove = true;
    ppp = e.Location;
}

private void pb1_MouseUp(object sender, MouseEventArgs e)
{
    canMove = false;

    pb pic;
    pic = (pb)sender;

    pic.x = pic.Location.X - pictureBox1.Left + 2;
    pic.y = pic.Location.Y - pictureBox1.Top + 3;

    if ((pic.Top < 10) & (pic.Left < 10))
    {
        for (int j = 0; j < listBox1.Items.Count; j++)
        {
            if (((tmar)listBox1.Items[j]).pbcs.Contains(pic))
                ((tmar)listBox1.Items[j]).pbcs.Remove(pic);
        }
        // pic.Remove(pic);
        this.Controls.Remove(pic);
    }
}

```

```
private void pb1_MouseMove(object sender, MouseEventArgs e)
{
    pb pic;
    if (canMove)
    {
        pic = (pb)sender;
        pic.Top += e.Y - ppp.Y;
        pic.Left += e.X - ppp.X;
    }
}
```

```
private void pb1_MouseDb1(object sender, MouseEventArgs e)
{
    pb pic;
    pic = (pb)sender;
    //  MessageBox.Show("The calculations are complete", "My Application",
    MessageBoxButtons.OKCancel, MessageBoxIcon.Asterisk);
    Form3 fm = new Form3();

    //  fm.x = (pic.Location.X - pictureBox1.Left + 4) * pictureBox1.Image.Width /
    pictureBox1.Width;
    //  fm.y = (pic.Location.Y - pictureBox1.Top + 4) * pictureBox1.Image.Height /
    pictureBox1.Height;
    fm.x = pic.Location.X - pictureBox1.Left+2;
    fm.y = pic.Location.Y - pictureBox1.Top+3;
    fm.pl = pic.lp;
    fm.pt = pic.tp;
    //  fm.setxy(cx, cy);
```

```

        if (fm.ShowDialog(this) == DialogResult.OK)
        {
            // Read the contents of testDialog's TextBox.
            //fm.TextBox1.Text="";
            // pic.Left = (Int16)((fm.x * pictureBox1.Width / pictureBox1.Image.Width) -
3 + pictureBox1.Left);
            // pic.Top = (Int16)((fm.y * pictureBox1.Height / pictureBox1.Image.Height)
- 3 + pictureBox1.Top);
            pic.Left = (Int16)(fm.x - 2 + pictureBox1.Left);
            pic.Top = (Int16)(fm.y - 3 + pictureBox1.Top);
            pic.x = fm.x;
            pic.y = fm.y;
            pic.lp = fm.pl;
            pic.tp = fm.pt;
        }

    }

private void button2_Click(object sender, EventArgs e)
{
    if (saveFileDialog1.ShowDialog() ==
System.Windows.Forms.DialogResult.OK)
    {
        XmlTextWriter textWriter = new
XmlTextWriter(saveFileDialog1.FileName, Encoding.UTF8);
        textWriter.WriteStartDocument();
        textWriter.WriteStartElement("head");
        textWriter.WriteEndElement();
        textWriter.Close();
    }
}

```

```

XmlDocument document = new XmlDocument();
document.Load(saveFileDialog1.FileName);
foreach (tmar mar in listBox1.Items)
{
    XmlNode element = document.CreateElement("marshrut");
    document.DocumentElement.AppendChild(element); // указываем
родителя
    XmlAttribute attribute = document.CreateAttribute("name"); // создаём
атрибут
    attribute.Value = mar.name; // устанавливаем значение атрибута
    element.Attributes.Append(attribute); // добавляем атрибут
    XmlAttribute at1 = document.CreateAttribute("port1"); // создаём атрибут
    at1.Value = mar.port1; // устанавливаем значение атрибута
    // at1.Value = "Стамбул";
    element.Attributes.Append(at1); // добавляем атрибут

    XmlAttribute at2 = document.CreateAttribute("port2"); // создаём атрибут
    at2.Value = mar.port2; // устанавливаем значение атрибута
    // at2.Value = "Одеса";
    element.Attributes.Append(at2); // добавляем атрибут

    foreach (pb pnt in mar.pbc)
    {
        XmlNode element1 = document.CreateElement("point");
        element.AppendChild(element1); // указываем родителя
        XmlAttribute attribute1 = document.CreateAttribute("x"); // создаём
атрибут
        attribute1.Value = pnt.x.ToString(); // устанавливаем значение
атрибута
        element1.Attributes.Append(attribute1); // добавляем атрибут

```

```

        XmlAttribute attribute2 = document.CreateAttribute("y"); // создаём
атрибут
        attribute2.Value = pnt.y.ToString(); // устанавливаем значение
атрибута
        element1.Attributes.Append(attribute2); // добавляем атрибут

        XmlAttribute attribute3 = document.CreateAttribute("lp"); // создаём
атрибут
        attribute3.Value = pnt.lp.ToString(); // устанавливаем значение
атрибута
        element1.Attributes.Append(attribute3); // добавляем атрибут
        XmlAttribute attribute4 = document.CreateAttribute("tp"); // создаём
атрибут
        attribute4.Value = pnt.tp.ToString(); // устанавливаем значение
атрибута
        element1.Attributes.Append(attribute4); // добавляем атрибут
    }

    document.Save(saveFileDialog1.FileName);
}
}
}

private void button3_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() ==
System.Windows.Forms.DialogResult.OK)
    {
        foreach (tmar mar in listBox1.Items)
        {

```

```

        mar.pbcs.Clear();
    }
    listBox1.Items.Clear();
    pbcnt = 0;
    Random rnd=new Random();

    XmlDocument xDoc = new XmlDocument();
    xDoc.Load(openFileDialog1.FileName);
    // получим корневой элемент
    XmlElement xRoot = xDoc.DocumentElement;
    // обход всех узлов в корневом элементе
    foreach (XmlNode xnode in xRoot)
    {
        // получаем атрибут name
        if (xnode.Attributes.Count > 0)
        {
            XmlNode attr = xnode.Attributes.GetNamedItem("name");
            XmlNode at1 = xnode.Attributes.GetNamedItem("port1");
            XmlNode at2 = xnode.Attributes.GetNamedItem("port2");

            tmar tm = new tmar { name = attr.Value, port1=at1.Value,
port2=at2.Value };
            tm.pbcs = new List<pb>();
            int c1 = rnd.Next(255);
            int c2 = rnd.Next(255);
            int c3 = rnd.Next(255);

            foreach (XmlNode xnode1 in xnode)
            {

```

```

if (xnode1.Attributes.Count > 0)
{
    XmlNode attr1 = xnode1.Attributes.GetNamedItem("x");
    XmlNode attr2 = xnode1.Attributes.GetNamedItem("y");
    XmlNode attr3 = xnode1.Attributes.GetNamedItem("lp");
    XmlNode attr4 = xnode1.Attributes.GetNamedItem("tp");
    if ((attr2 != null) && (attr1 != null))
    {
        int xx = 0;
        if (Int32.TryParse(attr1.Value, out xx))
        {
            int yy = 0;
            if (Int32.TryParse(attr2.Value, out yy))
            {

                pb pb2;
                pb2 = new pb();

                // pb2.x = e.X + pictureBox1.Left - 4;
                // pb2.y = e.Y + pictureBox1.Top - 4;

                pb2.x = xx;
                pb2.y = yy;

                pb2.Location = new System.Drawing.Point(xx +
pictureBox1.Left - 2, yy + pictureBox1.Top - 3);
                pbcnt++;
                pb2.Name = "pb" + pbcnt.ToString();
                pb2.Size = new System.Drawing.Size(8, 8);
            }
        }
    }
}

```

```

        pb2.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage;
        pb2.TabIndex = 1;
        pb2.TabStop = false;
        this.Controls.Add(pb2);
        pb2.BringToFront();
        pb2.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.pb1_MouseMove);
        pb2.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.pb1_MouseDown);
        pb2.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.pb1_MouseUp);
        pb2.MouseDoubleClick += new
System.Windows.Forms.MouseEventHandler(this.pb1_MouseDb1);

        Bitmap Bmp = new Bitmap(8, 8);
        using (Graphics gfx = Graphics.FromImage(Bmp))
        {
            using (SolidBrush brush = new
SolidBrush(System.Drawing.Color.FromArgb(192, 16, 16)))
            {
                gfx.FillRectangle(brush, 0, 0, 8, 8);
            }

            using (SolidBrush brush = new
SolidBrush(System.Drawing.Color.FromArgb(c1, c2, c3)))
            {
                gfx.FillRectangle(brush, 2, 2, 4, 4);
            }
        }
    }

```

```

        pb2.Image = Bmp;
        //          listBox2.Items.Add(pb2.x.ToString() + " " +
pb2.y.ToString());

        if ((attr3 != null) && (attr4 != null))
        {
            int blp = 0;
            if (Int32.TryParse(attr3.Value, out blp)) pb2.lp=blp ;
            int blt = 0;
            if (Int32.TryParse(attr4.Value, out blt)) pb2.tp = blt;
        }

        tm.pbcs.Add(pb2);
    }
}
}
}

}

listBox1.Items.Add(tm);
}
}
}

}

int[,] metr;

```

```

int nmar;

private void button4_Click(object sender, EventArgs e)
{
    // Sorting every mar
    /*  foreach (tmar mar in listBox1.Items)
        {
            // pb[] qmar = new pb[mar.pbc.Count];
            // for (int i = 0; i < mar.pbc.Count; i++) qmar[i] = mar.pbc[i];
            // IEnumerable<pb> sortp = qmar.OrderBy(x => x.tp);

            // var newlist = mar.pbc.OrderBy(x => x.tp).ToList();
            mar.pbc = mar.pbc.OrderBy(x => x.tp).ToList();
            // IEnumerable<tmar> quer = mar.pbc.OrderBy(x=>x.tp);
        } */

    //Init base variable
    nmar = listBox1.Items.Count;
    // listBox2.Items.Add(nmar.ToString());
    metr = new int [nmar,nmar];

    // aproximation
    foreach (tmar mar in listBox1.Items)
    {
        if (mar.pbc.Count>0)
            for (int i = 0; i < mar.pbc.Count - 1; i++)
            {
                mar.pbc[i].k = ((float)(mar.pbc[i + 1].y - mar.pbc[i].y)) /
                ((float)(mar.pbc[i + 1].x - mar.pbc[i].x));
                mar.pbc[i].b = mar.pbc[i].y - mar.pbc[i].k * mar.pbc[i].x;
            }
    }
}

```

```

    }
}

// Calc vidstani
for (int i = 0; i < nmar; i++)
    for (int j = 0; j < nmar; j++)
        if (i == j) metr[i, j] = 0;
        else
        {
            tmar m1=(tmar)listBox1.Items[i];
            tmar m2=(tmar)listBox1.Items[j];
            float r = 0;
            if (m2.pbc.Count>0)
                for (int l = 0; l < m2.pbc.Count; l++)
                    {
                        int x1 = m2.pbc[l].x;
                        int y1 = m2.pbc[l].y;
                        float min = 300000;
                        for (int m = 0; m < m1.pbc.Count-1; m++)
                            {
                                float k=m1.pbc[m].k;
                                float b=m1.pbc[m].b;
                                float r1=0;
                                int mx = (int)((k * y1 + x1 - k * b) / (1 + k * k));
                                int my = (int) (k * mx+b);
                                if (((mx > m1.pbc[m].x) & (mx < m1.pbc[m + 1].x)) | ((mx >
m1.pbc[m+1].x) & (mx < m1.pbc[m ].x)))
                                    {
                                        r1 = (float)Math.Sqrt(Math.Pow(mx - x1, 2) + Math.Pow(my -
y1, 2));

```

```

    }
    else
    {
        float r2 = (float)Math.Sqrt(Math.Pow(m1.pbc[m].x - x1, 2) +
Math.Pow(m1.pbc[m].y - y1, 2));
        float r3 = (float)Math.Sqrt(Math.Pow(m1.pbc[m+1].x - x1, 2) +
Math.Pow(m1.pbc[m+1].y - y1, 2));
        if (r2 < r3) r1 = r2; else r1 = r3;
    }
    if (r1 < min) min = r1;
}
r += min;
}
if (m2.pbc.Count > 0) r = r / m2.pbc.Count; else r = 30000;
metr[i, j] = (int)r;
}

```

// Zavershenna calc vidstani

// Vivid vidstanej

```

listBox2.Items.Add("Відстані");
for (int i = 0; i < nmar; i++)
{
    string s = "";
    for (int j = 0; j < nmar; j++) s += metr[i, j].ToString() + " ";
    listBox2.Items.Add(s);
}

```

// virivnuvanna

```

for (int i = 0; i < nmar; i++)

```

```

    for (int j = 0; j < nmar; j++)
        if (i != j)
            if (metr[i, j] < metr[j, i]) metr[j, i] = metr[i, j];

// Vivid vidstanej
listBox2.Items.Add("Відстані комутативні");
for (int i = 0; i < nmar; i++)
{
    string s = "";
    for (int j = 0; j < nmar; j++) s += String.Format("{0,4}", metr[i, j]);
    listBox2.Items.Add(s);
}
}

class tsmar
{
    public int ind { set; get; }
    public int okl { set; get; }
    public int sokl { set; get; }
}

List<List<int>> grup;

private void button5_Click(object sender, EventArgs e)
{
    List<tsmar> lp = new List<tsmar>();
    int por=int.Parse(textBox1.Text);
    for (int i=0; i<nmar; i++)
    {
        int cnt=0;

```

```

int sum = 0;
for (int j = 0; j < nmar; j++)
    if (i != j)
        if (metr[i, j] < por)
            {
                cnt++;
                sum += metr[i, j];
            }
lp.Add(new tsmar{ind=i, okl=cnt,sokl=sum});
}

// Sortuvanna po k-ti marshrutiv okolu (pri odnakov k-ti - po sumarnij vidstani)
List<tsmar> sp = lp.OrderByDescending(x =>
x.okl).ThenBy(y=>y.sokl).ToList();

foreach (var x in sp)
    listBox2.Items.Add((x.ind+1).ToString() + " " + x.okl.ToString()+" " +
x.sokl.ToString());

// Grupuvanna
grup = new List<List<int>>();
int ii = 0;
while ((ii<nmar) & (sp.Count>0))
{
    if (sp[0].okl > 0)
    {
        List<int> qq = new List<int>();
        qq.Add(sp[0].ind);

        for (int j = 0; j < nmar; j++)

```

```

        if (j != sp[0].ind)
            if (metr[sp[0].ind, j] < por)
                {
                    qq.Add(j);
                    sp.Remove(sp.Where(x => x.ind == j).First());
                }
            // sp.Remove(sp.Where(x => x.ind == sp[0].ind).First());
            grup.Add(qq);
        }
    sp.Remove(sp.Where(x => x.ind == sp[0].ind).First());
    ii++;
}

```

```

listBox2.Items.Add("Вивід груп");
foreach (List<int> gr in grup)
{
    string s = "";
    foreach (var x in gr) s += ((tmar)listBox1.Items[x]).name + " ";
    listBox2.Items.Add(s);
}

```

```

// maluvanna centraln marshrut
Random rnd = new Random();
Graphics g = pictureBox1.CreateGraphics();
foreach (List<int> gr in grup)
{
    int c1 = rnd.Next(255);
    int c2 = rnd.Next(255);
    int c3 = rnd.Next(255);
    tmar m = (tmar)listBox1.Items[gr[0]];
}

```

```

    for (int i = 0; i < m.pbcs.Count-1; i++)
    {
        g.DrawLine(new Pen(Brushes.Red, 3), new Point(m.pbcs[i].x,
m.pbcs[i].y), new Point(m.pbcs[i+1].x, m.pbcs[i+1].y));

    }
}
}

```

```

private void button6_Click(object sender, EventArgs e)
{
    if (saveFileDialog1.ShowDialog() ==
System.Windows.Forms.DialogResult.OK)
    {
        XmlTextWriter textWriter = new
XmlTextWriter(saveFileDialog1.FileName, Encoding.UTF8);
        textWriter.WriteStartDocument();
        textWriter.WriteStartElement("head");
        textWriter.WriteEndElement();
        textWriter.Close();
        XmlDocument document = new XmlDocument();
        document.Load(saveFileDialog1.FileName);
        foreach (List<int> gr in grup)
        {
            tmar mar = (tmar)listBox1.Items[gr[0]];
            XmlNode element = document.CreateElement("marshrut");
            document.DocumentElement.AppendChild(element); // указываем
родителя

```

```
XmlAttribute attribute = document.CreateAttribute("name"); // создаём  
атрибут
```

```
attribute.Value = mar.name; // устанавливаем значение атрибута  
element.Attributes.Append(attribute); // добавляем атрибут
```

```
XmlAttribute at1 = document.CreateAttribute("port1"); // создаём атрибут  
at1.Value = mar.port1; // устанавливаем значение атрибута  
// at1.Value = "Стамбул";  
element.Attributes.Append(at1); // добавляем атрибут
```

```
XmlAttribute at2 = document.CreateAttribute("port2"); // создаём атрибут  
at2.Value = mar.port2; // устанавливаем значение атрибута  
// at2.Value = "Одеса";  
element.Attributes.Append(at2); // добавляем атрибут
```

```
foreach (pb pnt in mar.pbc)
```

```
{
```

```
XmlNode element1 = document.CreateElement("point");  
element.AppendChild(element1); // указываем родителя
```

```
XmlAttribute attribute1 = document.CreateAttribute("x"); // создаём  
атрибут
```

```
attribute1.Value = pnt.x.ToString(); // устанавливаем значение  
атрибута
```

```
element1.Attributes.Append(attribute1); // добавляем атрибут
```

```
XmlAttribute attribute2 = document.CreateAttribute("y"); // создаём  
атрибут
```

```
attribute2.Value = pnt.y.ToString(); // устанавливаем значение  
атрибута
```

```
element1.Attributes.Append(attribute2); // добавляем атрибут
```



```
{
public partial class Form1 : Form
{
    public class pb : PictureBox
    {
        public int x, y;
        public int lp, tp;
        public float k, b;
    }

    public class tmar
    {
        public string name;
        public String port1;
        public String port2;
        public List<pb> pbc;
        public override string ToString()
        {
            return name;
        }
    }

    private int mx, my, pbcnt;
    private bool canMove;
    private Point ppp;

    List<tmar> etalon;
    List<pb> marshrut;

    public Form1()

```

```

{
    InitializeComponent();
    marshrut = new List<pb>();
    etalon = new List<tmar>();
}

```

```

private void pictureBox1_MouseDoubleClick(object sender, MouseEventArgs e)
{

    mx = e.X * pictureBox1.Image.Width / pictureBox1.Width;
    my = e.Y * pictureBox1.Image.Height / pictureBox1.Height;

    pb pb2;
    pb2 = new pb();

    pb2.x = e.X + pictureBox1.Left - 4;
    pb2.y = e.Y + pictureBox1.Top - 4;
    pb2.Location = new System.Drawing.Point(e.X + pictureBox1.Left - 4, e.Y +
pictureBox1.Top - 4);
    pbcnt++;
    pb2.Name = "pb" + pbcnt.ToString();
    pb2.Size = new System.Drawing.Size(8, 8);
    pb2.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage;
    pb2.TabIndex = 1;
    pb2.TabStop = false;
    this.Controls.Add(pb2);
    pb2.BringToFront();
    pb2.MouseMove +=
new
System.Windows.Forms.MouseEventHandler(this.pb1_MouseMove);

```

```

        pb2.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.pb1_MouseDown);
        pb2.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.pb1_MouseUp);
        pb2.MouseDoubleClick += new
System.Windows.Forms.MouseEventHandler(this.pb1_MouseDb1);

```

```

        Bitmap Bmp = new Bitmap(8, 8);
        using (Graphics gfx = Graphics.FromImage(Bmp))
        {
            using (SolidBrush brush = new
SolidBrush(System.Drawing.Color.FromArgb(192, 16, 16)))
            {
                gfx.FillRectangle(brush, 0, 0, 8, 8);
            }
            using (SolidBrush brush = new
SolidBrush(System.Drawing.Color.FromArgb(255, 228, 235)))
            {
                gfx.FillRectangle(brush, 2, 2, 4, 4);
            }
        }
        pb2.Image = Bmp;
        // listBox2.Items.Add(pb2.x.ToString() + " " + pb2.y.ToString());
        marshrut.Add(pb2);
        // mr.pbcs.Add(pb2);
        // pbcs.Add(pb2);
    }

```

```

private void pb1_MouseDown(object sender, MouseEventArgs e)

```

```
{
    canMove = true;
    ppp = e.Location;
}
```

```
private void pb1_MouseUp(object sender, MouseEventArgs e)
```

```
{
    canMove = false;

    pb pic;
    pic = (pb)sender;

    pic.x = pic.Location.X - pictureBox1.Left + 2;
    pic.y = pic.Location.Y - pictureBox1.Top + 3;

    if ((pic.Top < 10) & (pic.Left < 10))
    {
        if (marshrut.Contains(pic)) marshrut.Remove(pic);
        // pbc.Remove(pic);
        this.Controls.Remove(pic);
    }
}
```

```
private void pb1_MouseMove(object sender, MouseEventArgs e)
```

```
{
    pb pic;
    if (canMove)
    {
        pic = (pb)sender;
```

```
        pic.Top += e.Y - ppp.Y;
        pic.Left += e.X - ppp.X;
    }
}
```

```
private void pb1_MouseDb1(object sender, MouseEventArgs e)
{
    pb pic;
    pic = (pb)sender;
    //  MessageBox.Show("The calculations are complete", "My Application",
    MessageBoxButtons.OKCancel, MessageBoxIcon.Asterisk);
    Form2 fm = new Form2();

    // fm.x = (pic.Location.X - pictureBox1.Left + 4) * pictureBox1.Image.Width /
    pictureBox1.Width;
    // fm.y = (pic.Location.Y - pictureBox1.Top + 4) * pictureBox1.Image.Height /
    pictureBox1.Height;
    fm.x = pic.Location.X - pictureBox1.Left + 2;
    fm.y = pic.Location.Y - pictureBox1.Top + 3;
    fm.pl = pic.lp;
    fm.pt = pic.tp;
    // fm.setxy(cx, cy);
    if (fm.ShowDialog(this) == DialogResult.OK)
    {
        // Read the contents of testDialog's TextBox.
        //fm.TextBox1.Text="";
        //  pic.Left = (Int16)((fm.x * pictureBox1.Width / pictureBox1.Image.Width)
- 3 + pictureBox1.Left);
```

```

        // pic.Top = (Int16)((fm.y * pictureBox1.Height / pictureBox1.Image.Height)
- 3 + pictureBox1.Top);
        pic.Left = (Int16)(fm.x - 2 + pictureBox1.Left);
        pic.Top = (Int16)(fm.y - 3 + pictureBox1.Top);
        pic.x = fm.x;
        pic.y = fm.y;
        pic.lp = fm.pl;
        pic.tp = fm.pt;
    }

```

```

}

```

```

private void button1_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() ==
System.Windows.Forms.DialogResult.OK)
    {
        if (etalon.Count > 0) etalon.Clear();

        XmlDocument xDoc = new XmlDocument();
        xDoc.Load(openFileDialog1.FileName);
        // получим корневой элемент
        XmlElement xRoot = xDoc.DocumentElement;
        // обход всех узлов в корневом элементе
        foreach (XmlNode xnode in xRoot)
        {
            // получаем атрибут name
            if (xnode.Attributes.Count > 0)
            {

```

```

XmlNode attr = xnode.Attributes.GetNamedItem("name");
XmlNode att1 = xnode.Attributes.GetNamedItem("port1");
XmlNode att2 = xnode.Attributes.GetNamedItem("port2");

tmar tm = new tmar { name = attr.Value, port1=att1.Value,
port2=att2.Value };
tm.pbcs = new List<pb>();

foreach (XmlNode xnode1 in xnode)
{
    if (xnode1.Attributes.Count > 0)
    {
        XmlNode attr1 = xnode1.Attributes.GetNamedItem("x");
        XmlNode attr2 = xnode1.Attributes.GetNamedItem("y");
        XmlNode attr3 = xnode1.Attributes.GetNamedItem("lp");
        XmlNode attr4 = xnode1.Attributes.GetNamedItem("tp");
        if ((attr2 != null) && (attr1 != null))
        {
            int xx = 0;
            if (Int32.TryParse(attr1.Value, out xx))
            {
                int yy = 0;
                if (Int32.TryParse(attr2.Value, out yy))
                {

                    pb pb2;
                    pb2 = new pb();

                    // pb2.x = e.X + pictureBox1.Left - 4;
                    // pb2.y = e.Y + pictureBox1.Top - 4;
                }
            }
        }
    }
}

```

```

        pb2.x = xx;
        pb2.y = yy;

        pb2.Location = new System.Drawing.Point(xx +
pictureBox1.Left - 2, yy + pictureBox1.Top - 3);
        pbcnt++;
        pb2.Name = "pb" + pbcnt.ToString();
        pb2.Size = new System.Drawing.Size(8, 8);
        pb2.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage;
        pb2.TabIndex = 1;
        pb2.TabStop = false;
        // this.Controls.Add(pb2);
        // pb2.BringToFront();
        pb2.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.pb1_MouseMove);
        pb2.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.pb1_MouseDown);
        pb2.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.pb1_MouseUp);
        pb2.MouseDoubleClick += new
System.Windows.Forms.MouseEventHandler(this.pb1_MouseDb1);

        Bitmap Bmp = new Bitmap(8, 8);
        using (Graphics gfx = Graphics.FromImage(Bmp))
        {
            using (SolidBrush brush = new
SolidBrush(System.Drawing.Color.FromArgb(192, 16, 16)))
            {

```



```

        etalon.Add(tm);
    }
}
listBox1.Items.Add(etalon.Count.ToString() + " марш.");

Graphics g = pictureBox1.CreateGraphics();
foreach (tmar m in etalon)
{

    for (int i = 0; i < m.pbc.Count - 1; i++)
    {
        g.DrawLine(new Pen(Brushes.Red, 3), new Point(m.pbc[i].x,
m.pbc[i].y), new Point(m.pbc[i + 1].x, m.pbc[i + 1].y));
    }
}
}
}

```

```

public static Int32 DamerauLevenshteinDistance(String source, String target)
{
    if (String.IsNullOrEmpty(source))
    {
        if (String.IsNullOrEmpty(target))
        {
            return 0;
        }
        else
        {
            return target.Length;
        }
    }
}

```

```
    }  
}  
else if (String.IsNullOrEmpty(target))  
{  
    return source.Length;  
}
```

```
Int32 m = source.Length;  
Int32 n = target.Length;  
Int32[,] H = new Int32[m + 2, n + 2];
```

```
Int32 INF = m + n;  
H[0, 0] = INF;  
for (Int32 i = 0; i <= m; i++) { H[i + 1, 1] = i; H[i + 1, 0] = INF; }  
for (Int32 j = 0; j <= n; j++) { H[1, j + 1] = j; H[0, j + 1] = INF; }
```

```
SortedDictionary<Char, Int32> sd = new SortedDictionary<Char, Int32>();  
foreach (Char Letter in (source + target))  
{  
    if (!sd.ContainsKey(Letter))  
        sd.Add(Letter, 0);  
}
```

```
for (Int32 i = 1; i <= m; i++)  
{  
    Int32 DB = 0;  
    for (Int32 j = 1; j <= n; j++)  
    {  
        Int32 i1 = sd[target[j - 1]];  
        Int32 j1 = DB;
```

```

        if (source[i - 1] == target[j - 1])
        {
            H[i + 1, j + 1] = H[i, j];
            DB = j;
        }
        else
        {
            H[i + 1, j + 1] = Math.Min(H[i, j], Math.Min(H[i + 1, j], H[i, j + 1])) +
1;
        }

        H[i + 1, j + 1] = Math.Min(H[i + 1, j + 1], H[i1, j1] + (i - i1 - 1) + 1 + (j -
j1 - 1));
    }

    sd[source[i - 1]] = i;
}

return H[m + 1, n + 1];
}

```

```

public int strdist(String source, String target)
{
    int l1 = source.Length;
    int l2 = target.Length;
    int dist = DamerauLevenshteinDistance(source, target);
    int max;
    if (l1 > l2) max = l1; else max = l2;
    if (max==0) max=1;
}

```

```
    return (dist * 100 / max);  
}
```

```
int[] metr;  
int[] maxv;  
int nmar;  
private void button2_Click(object sender, EventArgs e)  
{  
    //  
    metr = new int[etalon.Count];  
    nmar = etalon.Count;  
    maxv = new int[nmar];  
  
    // aproximation  
    foreach (tmar mar in etalon)  
    {  
        if (mar.pbc.Count > 0)  
            for (int i = 0; i < mar.pbc.Count - 1; i++)  
            {  
                mar.pbc[i].k = ((float)(mar.pbc[i + 1].y - mar.pbc[i].y)) /  
                ((float)(mar.pbc[i + 1].x - mar.pbc[i].x));  
                mar.pbc[i].b = mar.pbc[i].y - mar.pbc[i].k * mar.pbc[i].x;  
            }  
    }  
  
    // Calc vidstani  
    for (int j = 0; j < nmar; j++)  
    {  
        tmar m1 = etalon[j];
```

```

// tmar m2 = (tmar)listBox1.Items[j];
float r = 0;
maxv[j] = 0;
if (marshrut.Count > 0)
    for (int l = 0; l < marshrut.Count; l++)
        {
            int x1 = marshrut[l].x;
            int y1 = marshrut[l].y;
            // listBox1.Items.Add("x1=" + x1.ToString());
            // listBox1.Items.Add("y1=" + y1.ToString());
            float min = 300000;
            for (int m = 0; m < m1.pbcx.Count - 1; m++)
                {
                    float k = m1.pbcx[m].k;
                    float b = m1.pbcx[m].b;
                    // listBox1.Items.Add("k=" + k.ToString());
                    // listBox1.Items.Add("b=" + b.ToString());
                    float r1 = 0;
                    int mx = (int)((k * y1 + x1 - k * b) / (1 + k * k));
                    int my = (int)(k * mx + b);
                    // listBox1.Items.Add("mx=" + mx.ToString());
                    // listBox1.Items.Add("my=" + my.ToString());
                    // listBox1.Items.Add("xxx=" + m1.pbcx[m].x.ToString() + " " +
m1.pbcx[m+1].x.ToString());

                    if (((mx > m1.pbcx[m].x) & (mx < m1.pbcx[m + 1].x)) | ((mx >
m1.pbcx[m+1].x) & (mx < m1.pbcx[m].x)))
                        {

```

```

        r1 = (float)Math.Sqrt(Math.Pow(mx - x1, 2) + Math.Pow(my -
y1, 2));
    }
    else
    {
        float r2 = (float)Math.Sqrt(Math.Pow(m1.pbc[m].x - x1, 2) +
Math.Pow(m1.pbc[m].y - y1, 2));
        float r3 = (float)Math.Sqrt(Math.Pow(m1.pbc[m + 1].x - x1,
2) + Math.Pow(m1.pbc[m + 1].y - y1, 2));
        if (r2 < r3) r1 = r2; else r1 = r3;
    }
    // listBox1.Items.Add("r=" + r1.ToString());
    if (r1 < min) min = r1;
}
r += min;
if (min > maxv[j]) maxv[j] = (int) min;
}
if (marshrut.Count > 0) r = r / marshrut.Count; else r = 30000;
metr[j] = (int)r;
}
// Zavershenna calc vidstani

// Vivid vidstanej
listBox1.Items.Add("Відстані");
for (int j = 0; j < nmar; j++) listBox1.Items.Add(etalon[j].name+" "+ metr[
j].ToString());

int por = int.Parse(textBox1.Text);
int rmin = metr[0];

```

```

int irmin = 0;
for (int i=1;i<nmar;i++)
    if (metr[i] < rmin)
    {
        rmin = metr[i];
        irmin = i;
    }
if (rmin <= por)
{
    listBox1.Items.Add("Класифіковано");
    listBox1.Items.Add(etalon[irmin].name);

    bool oz1;
    if (((strdist(textBox2.Text, etalon[irmin].port1) < 15) &
(strdist(textBox3.Text, etalon[irmin].port2) < 15))|
        ((strdist(textBox2.Text, etalon[irmin].port2) < 15) &
(strdist(textBox3.Text, etalon[irmin].port1) < 15)) ) oz1=true; else oz1=false;
    if (oz1) listBox1.Items.Add("Опису відповідає"); else
listBox1.Items.Add("Опису невідповідає");

    // listBox1.Items.Add(textBox2.Text);
    // listBox1.Items.Add(etalon[irmin].port1);

    // listBox1.Items.Add("text dist " + strdist(textBox2.Text,
etalon[irmin].port1));

    listBox1.Items.Add("Відстань "+metr[irmin]);

```

```
listBox1.Items.Add("Мах відх. " + maxv[irmin]);
```

```
listBox1.Items.Add("Рівень загроз");
```

```
if (oz1 & (rmin < (0.5 * por)) & (maxv[irmin]< (0.8*por)))
```

```
listBox1.Items.Add("Низький");
```

```
else if (!oz1 | (oz1&((rmin > (0.75 * por)) | ((maxv[irmin] > (0.9*por))))))
```

```
listBox1.Items.Add("Високий");
```

```
else listBox1.Items.Add("Середній");
```

```
}
```

```
else listBox1.Items.Add("Не класифіковано");
```

# ДОДАТОК Б

(обов'язковий)

## КОПІЯ ТЕЗ

International scientific journal «Computer systems and information technologies»

УДК 004.94

DOI: 10.31891/CSIT-2020-2-6

БОРОВИК О. В.

Національна академія Державної прикордонної служби України ім. Богдана Хмельницького, м. Хмельницький

БОРОВИК Д. О., КОСТЕЛЬНА Т. В.

Хмельницький національний університет

### ЩОДО НЕОБХІДНОСТІ УДОСКОНАЛЕННЯ МЕТОДУ КЛАСТЕРИЗАЦІЇ МАРШРУТІВ СУДЕН ЯК ПРОЦЕДУРНОГО МОДУЛЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ОБРОБКИ ДАНИХ У СИСТЕМІ ВІСВІТЛЕННЯ НАДВОДНОЇ ОБСТАНОВКИ

*Стаття присвячена аналізу існуючого методу кластеризації маршрутів суден в межах виключної морської (економічної) зони з позиції обґрунтування можливості його застосування або визначення шляхів його удосконалення для використання в системі висвітлення надводної обстановки в інтересах забезпечення національної безпеки на державному кордоні.*

*У результаті проведеного дослідження встановлено, що існуючий метод кластеризації маршрутів суден не може у якому вигляді бути застосованим для використання в системі висвітлення надводної обстановки як процедурний модуль автоматизованої системи обробки даних, який би забезпечував достатній рівень достовірності виявлення ознак порушення прикордонного законодавства в межах виключної морської (економічної) зони. Також встановлено, що удосконалення методу кластеризації маршрутів суден має стосуватися наступних питань: визначення таких видів апроксимації для формування неперервних маршрутів руху окремих суден, які б забезпечували достатній рівень достовірності результатів на заданих експериментальних наборах даних, що характеризують координати дискретних точок місць знаходження суден в окремі моменти часу; обґрунтування вибору такої кількості експериментальних точок місць знаходження суден і безпосередньо комбінації точок, які б забезпечили достатній рівень достовірності висновків, що формуються; дослідження різних метрик для встановлення подібності довільних маршрутів руху суден з позиції забезпечення однозначності висновків; адаптації запропонованого інструментарію на граничні початкові умови досліджуваної задачі; адаптації методу для випадку довільної складності маршруту руху суден; формування такого методу побудови еталонного маршруту у межах шуканого кластеру, який би забезпечував відтворення маршруту у вигляді неперервної траєкторії.*

*Ключові слова: система висвітлення надводної обстановки, процедурний модуль, автоматизована система обробки даних, метод, кластеризація, маршрут.*

BOROVYK O.

The National Academy of the State Border Guard Service of Ukraine named after Bohdan Khmelnytskyi

Khmelnytskyi city

BOROVYK D., KOSTELNA T.

Khmelnytskyi National University

### ON THE NECESSITY OF IMPROVING THE METHOD OF CLUSTERIZATION OF VESSEL ROUTES AS A PROCEDURAL MODULE OF THE AUTOMATED DATA PRODUCTION PROCESSING SYSTEM IN SUN

*The article is devoted to the analysis of the existing method of clustering of ship routes within the exclusive sea (economic) zone from the standpoint of substantiation of its application or identification of ways to improve it for use in the surface lighting system in the interests of national security at the state border.*

*The study found that the existing method of clustering ship routes can not be explicitly applicable for use in the surface lighting system as a procedural module of the automated data processing system, which would provide a sufficient level of reliability to detect signs of violations of border legislation within the exclusive maritime (economic) zone. It is also established that the improvement of the method of clustering of routes of ships should address the following issues: definition of such types of approximation for the formation of continuous routes of individual ships, which would provide a sufficient level of reliability of the results on given experimental data sets; time; justification for the choice of such a number of experimental points of location of ships and directly a combination of points that would ensure a sufficient level of reliability of the conclusions formed; study of different metrics to establish the similarity of arbitrary routes of ships from the standpoint of ensuring unambiguous conclusions; adaptation of the proposed tools to the boundary initial conditions of the studied problem; adaptation of the method for the case of arbitrary complexity of the route of vessels; formation of such a method of constructing a reference route within the desired cluster, which would ensure the reproduction of the route in the form of a continuous trajectory.*

*Keywords: surface lighting system, procedural module, automated data processing system, method, clustering, route.*

**Постановка проблеми.** Важливим чинником стабільності держави є надійна охорона кордону. В сучасних умовах постійного зростання обсягів даних, які необхідно враховувати при прийнятті рішень, надзвичайно важливою є інформаційно-телекомунікаційна складова сучасної моделі охорони кордону. Ця складова реалізується з використанням інтегрованої інформаційно-телекомунікаційної системи (ІТС) «Гарт». Однією з важливих складових ІТС «Гарт» є інформаційно-телекомунікаційна система (ІТС) Морської охорони «Гарт-12».

На сьогодні «Гарт-12» використовується лише для висвітлення надводної обстановки в Азово-Чорноморській акваторії. При цьому, існує можливість отримання інформації з ІТС «Гарт-12» про поточне положення суден, порт відправки та порт призначення, тип вантажу та інші додаткові дані. При виборі судна у системі висвітлення надводної обстановки (СВНО) існує можливість перегляду попередніх точок його маршруту, отриманих в певні дискретні моменти часу. При цьому, візуалізація маршруту здійснюється з використанням лінійної апроксимації (наявні точки з'єднуються відрізками прямих) [1].

Слід відмітити, що по такій, наявній у межах СВНО інформації, навіть експертам може бути важко визначити потенційні загрози, що стосуються порушення прикордонного законодавства. Оскільки ж відхилення судна від еталонного маршруту в окремих випадках може являти собою порушення правил прикордонного режиму, то актуальним завданням є ідентифікація тих ознак, які можуть вказувати на зазначене порушення.

Виявлення цих ознак є достатньо складним завданням, оскільки пов'язане з необхідністю обробки значних обсягів даних щодо надводної обстановки в ІТС «Гарт-12» та складністю аналізу ризиків. Саме цим пояснюється той факт, що для вирішення цього завдання авторами роботи [1] запропоновано підхід, що стосується створення відповідної автоматизованої системи обробки даних, яка б містила процедури кластеризації маршрутів, встановлення еталонних маршрутів руху суден в СВНО, виявлення аномалій руху суден і оцінки ризиків. Зрозуміло, що від ефективності функціонування цієї системи безпосередньо залежить ефективність СВНО.

Разом з тим, незважаючи на наявність підходів до комплексної оцінки ефективності функціонування СВНО, загалом [2], на сьогодні ще залишається невирішеним питання оцінки ефективності автоматизованої системи обробки даних. Оскільки остання є інтелектуальною складовою СВНО, то питання оцінки її ефективності є особливо актуальним завданням, яке потребує вирішення.

**Аналіз останніх досліджень і публікацій.** Останнє передбачає, насамперед, проведення аналізу науково-методичних підходів до розбудови вказаної автоматизованої системи обробки даних.

Відповідні питання знайшли відображення в ряді наукових праць, зокрема [1–6]. Так, у роботі [1] розглянуті можливі підходи до просторового аналізу даних про маршрути суден в ІТС Морської охорони «Гарт-12». Показано доцільність введення особливої метрики для класифікації маршрутів, які представлені в ІТС «Гарт-12» впорядкованими множинами точок, що отримані в довільні моменти часу. Для випадку лінійної апроксимації даних отримано вирази для встановлення ступеню подібності маршрутів. Однак питанням адекватності запропонованої метрики у цій праці увага приділена не належна.

Логічним продовженням вказаного дослідження стали формалізація та алгоритмізація методики визначення відстаней між маршрутами суден в «Гарт-12». Відповідні питання знайшли відображення у працях [3, 4]. Так, у роботі [3] деталізована методика обчислення метрики для встановлення міри подібності маршрутів суден на основі даних в ІТС морської охорони «Гарт-12». Перевірено комутативність цієї метрики. З метою класифікації маршрутів розроблено метод кластеризації, який дозволяє встановивши належність даних про окремі маршрути до кластерів, визначити еталонний маршрут. Подібний підхід досліджувався і в праці [4]. Проте питання адекватності методики перевірки комутативності метрики у цих працях увага не приділена взагалі.

Наявність методики кластеризації наявних даних про маршрути суден дозволила авторам роботи [5] запропонувати підхід до визначення просторових аномалій руху суден та оцінки ступеня ризиків порушення прикордонного законодавства. На основі кластеризації було запропоновано технологію проведення класифікації наявних даних у базі ІТС «Гарт-12» та визначення еталонних маршрутів. Дані про ці маршрути дозволяють класифікувати нові дані, які надходять до СВНО, та проводити на їх основі аналіз ризиків. Також у роботі визначено логічні вирази для встановлення рівня загроз щодо порушення прикордонного законодавства на морській ділянці кордону.

Таким чином, з проведеного аналізу випливає, що базовим положенням, яке визначає ефективність автоматизованої системи обробки даних, є метод кластеризації маршруту суден. Саме цей метод є інтелектуальною основою можливої виявлення ознак порушення прикордонного законодавства. А отже, саме достовірність результату, отриманого за допомогою цього методу, здійснює визначальний вплив на ефективність досліджуваної автоматизованої системи.

Цим і визначається необхідність проведення оцінки достовірності результатів, які формуються методом кластеризації маршрутів суден.

**Метою статті** є аналіз методу кластеризації маршрутів суден з позиції обґрунтування можливості його застосування або визначення шляхів його удосконалення для використання в СВНО як процедурного модуля автоматизованої системи обробки даних, який би забезпечував достатній рівень достовірності виявлення ознак порушення прикордонного законодавства.

**Виклад основного матеріалу дослідження.** У базі даних ІТС морської охорони «Гарт-12» інформація про пересування суден зберігається у вигляді впорядкованої послідовності точок з координатами суден у певні моменти часу.

Уявлення про місце знаходження суден у фіксований момент часу можна сформулювати, наприклад, з використанням рис. 1.

Сукупність координат точок місцезнаходження  $i$ -го судна можна позначити впорядкованою множиною (кортежем)

$$M_i = \{(x_1, y_1), \dots, (x_{k_i}, y_{k_i})\}, \quad (1)$$

де  $k_i$  – потужність множини  $M_i$ .

Слід зауважити, що потужності відповідних множин для різних суден, у загальному випадку, різні. Причому потужності цих множин можуть відрізнятися достатньо суттєво навіть у випадку, якщо судна



$$x = \frac{y_j \cdot k_i + x_j - b_i \cdot k_i}{1 + k_i^2}, \quad (3)$$

$$y = k_i \cdot x + b_i,$$

де  $k_i, b_i$  – параметри лінійного рівняння прямої, що описує  $i$ -й сегмент, який з'єднує точки  $(x_i, y_i)$  і  $(x_{i+1}, y_{i+1})$  множини  $M_1$ . Вказані параметри можуть бути нескладно знайдені з використанням рівняння прямої, що проходить через дві задані точки.

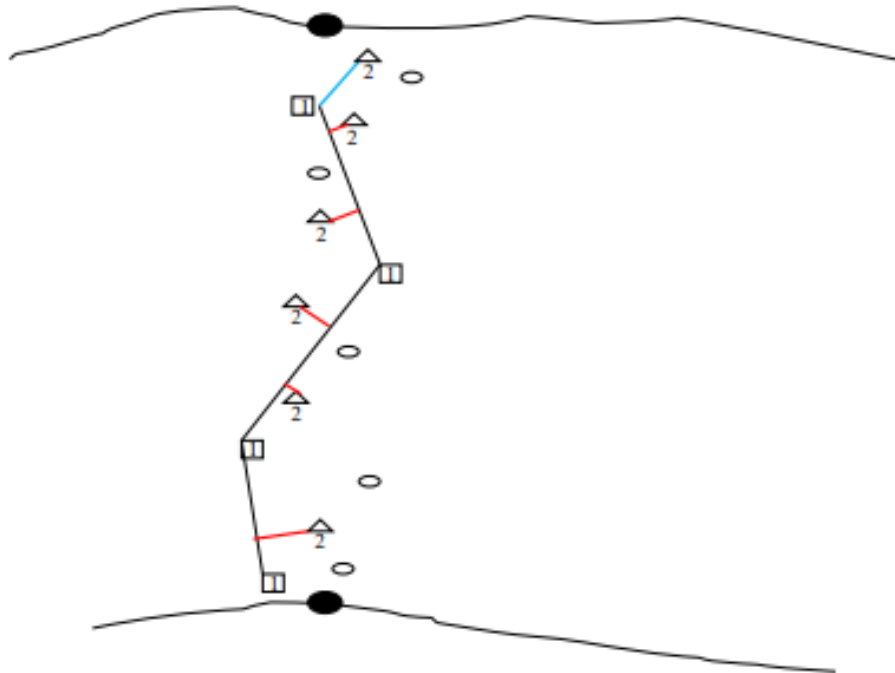


Рис.2. Підхід до визначення ступеня подібності маршрутів  $M_1$  та  $M_2$  за допомогою метрики  $R(M_1, M_2)$

Тоді

$$R_{\min j} = \sqrt{\left(x_j - \frac{y_j \cdot k_i + x_j - b_i \cdot k_i}{1 + k_i^2}\right)^2 + \left(y_j - k_i \cdot \frac{y_j \cdot k_i + x_j - b_i \cdot k_i}{1 + k_i^2} - b_i\right)^2}. \quad (4)$$

Формула (4) може застосовуватись у випадку, коли найкоротша відстань від  $j$ -ї точки множини  $M_2$  до встановленого першого маршруту знаходиться на перпендикулярі від  $j$ -ї точки множини  $M_2$  до деякого сегмента неперервного апроксимованого першого маршруту (на рис. 2 це відрізки, що зображені червоним кольором). Однак, можливий випадок, коли величина  $x$ , що отримана з (3), може знаходитись поза діапазоном  $[x_i, x_{i+1}]$ . Тоді величину  $R_{\min j}$  пропонується знаходити як найменшу відстань від  $j$ -ї точки множини  $M_2$  до кінців відрізка  $i$ -го сегменту (тобто до точок  $(x_i, y_i)$  і  $(x_{i+1}, y_{i+1})$  множини  $M_1$ ) (на рис. 2 це відрізок, що зображений синім кольором). Тоді, з урахуванням цього, у загальному випадку:

$$R_{\min j} = \begin{cases} \sqrt{\left(x_j - \frac{y_j \cdot k_i + x_j - b_i \cdot k_i}{1 + k_i^2}\right)^2 + \left(y_j - k_i \cdot \frac{y_j \cdot k_i + x_j - b_i \cdot k_i}{1 + k_i^2} - b_i\right)^2}, & x \in [x_i, x_{i+1}], \\ \min \left\{ \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}, \sqrt{(x_j - x_{i+1})^2 + (y_j - y_{i+1})^2} \right\}, & x \notin [x_i, x_{i+1}]. \end{cases} \quad (5)$$

Послідовне використання виразів (5), (3), (2) визначає методику обчислення метрики  $R(M_1, M_2)$ .

Слід зауважити, що величина метрики  $R$ , яка використовується для оцінки ступеня подібності першого та другого маршрутів залежить від порядку слідування аргументів у записі для  $R$ . Так, якщо в записі  $R(M_1, M_2)$  аргументи  $M_1$  та  $M_2$  поміняти місцями, то запис метрики матиме вигляд  $R(M_2, M_1)$ , а її відшукання для того ж випадку початкових даних, що й наведені на рис. 2, може бути оцінене з рис. 3.

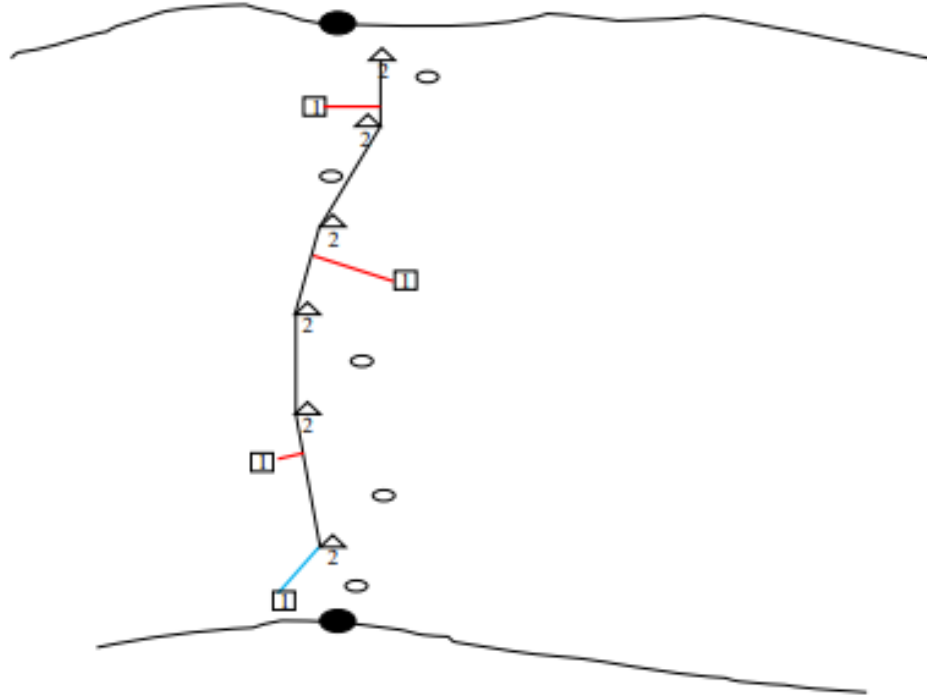


Рис. 3. Підхід до визначення ступеня подібності маршрутів  $M_2$  та  $M_1$  за допомогою метрики  $R(M_2, M_1)$

Для розрахунку  $R(M_2, M_1)$  необхідно насамперед здійснити встановлення (відновлення, генерування) неперервного маршруту руху другого судна між заданими у дискретній множині  $M_2$  точками з використанням лінійної апроксимації (див. рис 3). Тоді величина  $R(M_2, M_1)$  визначається, як середнє відхилення точок множини  $M_1$ , які описують дискретний маршрут руху першого судна, від апроксимованого неперервного другого маршруту у вигляді:

$$R(M_2, M_1) = \frac{\sum_{j=1}^{k_1} R_{\min_j}}{k_1}, \quad (6)$$

де  $R_{\min_j}$  є найкоротшою відстанню від  $j$ -ї точки множини  $M_1$  до встановленого другого маршруту, тобто до найближчого деякого  $i$ -го сегменту прямої, який з'єднує точки  $(x, y)$  і  $(x_{i+1}, y_{i+1})$  множини  $M_2$ , а  $k_1$  – кількість досліджуваних точок першого маршруту.

При використанні лінійної апроксимації сегментів другого маршруту координати  $(x, y)$  точки  $i$ -го сегменту прямої, який з'єднує точки  $(x, y)$  і  $(x_{i+1}, y_{i+1})$  другого маршруту, яка є найближчою до  $j$ -ї точки множини  $M_1$ , координати якої рівні  $(x_j, y_j)$ , також можна отримати за допомогою формул (3), але з урахуванням того, що  $k_i, b_i$  – параметри лінійного рівняння прямої, що описує  $i$ -й сегмент, який з'єднує

точки  $(x, y)$  і  $(x_{i+1}, y_{i+1})$  множини  $M_2$ . Вказані параметри можуть бути нескладно знайдені з використанням рівняння прямої, що проходить через дві задані точки. Тоді  $R_{\min j}$  можна визначити за допомогою формули (4) у випадку, коли найкоротша відстань від  $j$ -ї точки множини  $M_1$  до встановленого другого маршруту знаходиться на перпендикулярі від  $j$ -ї точки множини  $M_1$  до деякого сегмента неперервного апроксимованого другого маршруту (на рис. 3 це відрізки, що зображені червоним кольором).

У випадку ж, коли величина  $x$ , що отримана з (3), знаходиться поза діапазоном  $[x_i, x_{i+1}]$ , величину  $R_{\min j}$  пропонується знаходити як найменшу відстань від  $j$ -ї точки множини  $M_1$  до кінця відрізка  $i$ -го сегменту (тобто до точок  $(x, y)$  і  $(x_{i+1}, y_{i+1})$  множини  $M_2$ ) (на рис. 3 це відрізок, що зображений синім кольором). Тоді, з урахуванням цього, у загальному випадку  $R_{\min j}$  буде також знаходитися за допомогою формули (5).

Аналіз підходів і даних, що наведені на рис. 2, 3, спонукає до необхідності перевірки рівності метрик  $R(M_1, M_2)$ ,  $R(M_2, M_1)$ . Адже, для однієї і тієї ж пари маршрутів у залежності від обраної метрики у разі неспівпадання  $R(M_1, M_2)$  і  $R(M_2, M_1)$  можуть мати місце різні висновки щодо подібності чи неподібності маршрутів. Саме тому для перевірки рівності величин  $R(M_1, M_2)$  і  $R(M_2, M_1)$  у роботі [1] було проведено окреме дослідження для вихідних даних, які наведені на рис. 4.

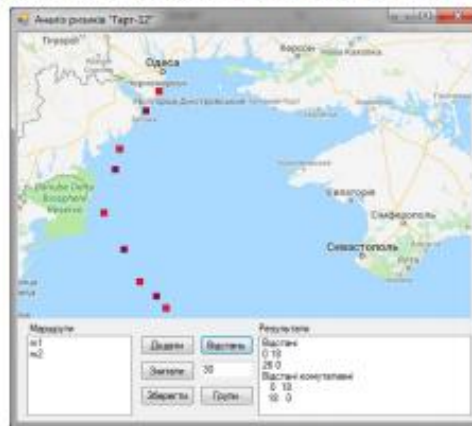


Рис. 4. Розрахунок метрик  $R(M_1, M_2)$  і  $R(M_2, M_1)$

Застосування формул (2) – (6) для початкових даних рис. 4 дозволило отримати, що  $R(M_1, M_2) = 18$ ,  $R(M_2, M_1) = 26$ . А отже, у загальному випадку, дійсно  $R(M_1, M_2) \neq R(M_2, M_1)$ .

Відсутність властивості комутативності щодо метрик  $R(M_1, M_2)$  і  $R(M_2, M_1)$  спонукала авторів роботи [1] до формування пропозицій щодо застосування нової метрики, яка б не містила недоліку, що стосувався б неоднозначності формування висновків відносно подібності маршрутів.

Ця пропозиція має вигляд метрики виду:

$$R^k(M_1, M_2) = \min(R(M_1, M_2), R(M_2, M_1)). \quad (7)$$

Для перевірки властивості комутативності для метрики (7) було проведено дослідження для початкових даних, що наведені на рис. 5. Результати дослідження, що описані у праці [3], вказують на комутативність метрики (7).

Однак питання об'єктивності результатів, отриманих за допомогою метрики (7), у роботі [3] увага не приділялася.

**Завдання 2.** Введення деякого граничного порогового рівня  $r_{pk}$  (даний рівень визначається на основі експертної оцінки) щодо величини відхилення маршрутів і його порівняння з метрикою (7) дозволяє робити висновки про ступінь подібності маршрутів  $M_1$  і  $M_2$  незалежно від того, для якого з маршрутів формується апроксимаційна неперервна траєкторія руху, і відносно якого здійснюється оцінка близькості до такої траєкторії.

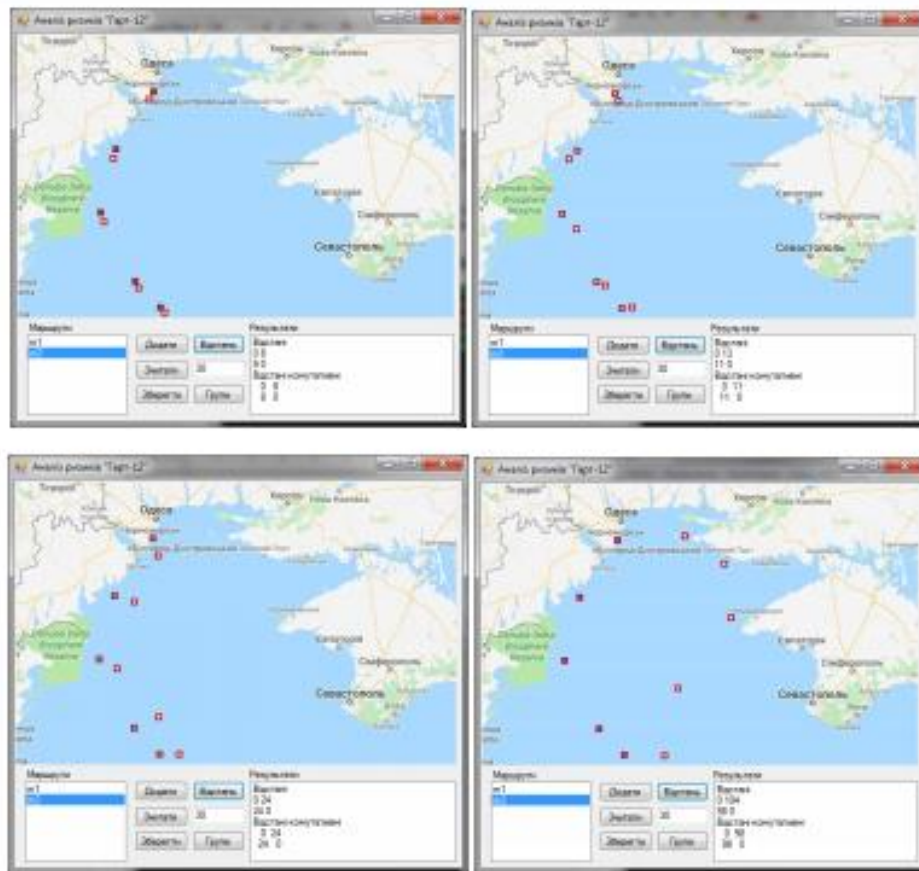


Рис. 5. Приклади обчислення метрики (7)

Встановлення геометричної близькості маршрутів за допомогою формули (7) дає можливість провести їх подальшу класифікацію та поділ на кластери (кластеризацію) з подальшим визначенням у кожному з них еталонного маршруту. Еталонні маршрути можуть використовуватись для оцінки ступеня відхилення судна, відносно якого проводиться аналіз. Також може здійснюватись подальша класифікація нових даних, що поступають до системи, та оновлення або уточнення меж кластера.

Для проведення кластеризації маршрутів на основі наявних у базі «Гарт-12» даних у роботі [3] пропонується побудувати матрицю  $\mathbf{R}$ , елементи  $r_{ij}$  якої визначаються наступним чином:

$$r_{ij} = \begin{cases} 0, & i = j, \\ R^k(M_i, M_j), & i \neq j, \end{cases} \quad (8)$$

де  $M_i, M_j$  – маршрути відповідно  $i$ - та  $j$ -го суден.

Після формування матриці  $\mathbf{R}$  для кожного її рядка, тобто для  $i = \text{const}$ , обчислюється:

– кількість  $n_i$  елементів, які не перевищують порогового для кластеризації рівня  $r_{pk}$ , тобто кількість елементів, для яких у рядку  $i$  виконується нерівність  $R^k(M_i, M_j) \leq r_{pk}$ ;

– величина  $s_i = \sum R^k(M_i, M_j)$ , яка формується лише з тих значень  $R^k(M_i, M_j)$ , для яких виконується умова  $R^k(M_i, M_j) \leq r_{pk}$ .

На основі отриманих даних, які сортується за зменшенням  $n_i$  (для елементів з однаковими значеннями  $n_i$  здійснюється сортування за збільшенням  $s_i$ ), формується впорядкована множина  $\mathbf{M}_i$ , елементами якої є набори  $(i, n_i(i), s_i(i))$ .

Далі множина  $M_i$  використовується для визначення тих маршрутів, які можуть бути включені до одного і того ж кластера. Здійснюється це за допомогою наступного алгоритму:

1. Якщо для першого елемента множини  $M_i$  виконується умова  $n_d(i) > 0$ , то створюється кластер з центральним маршрутом  $i$  і в нього включаються всі маршрути  $j$ , для яких  $R^k(M_i, M_j) \leq r_{pk}$ .

2. З множини  $M_i$  видаляються всі набори  $(i, n_d(i), s_i(i))$ , в яких перший елемент  $i$  співпадає із значеннями  $j$ , які визначили маршрути, що увійшли до створеного кластера.

3. Кроки 1–2 алгоритму повторюються до тих пір, поки потужність множини  $M_i$  є більшою за 0.

Реалізація такого підходу дозволяє встановити всі кластери (коридори руху суден), які задовольняють умові не перевищення порогу  $r_{pk}$ .

Проведений вище аналіз задач 1, 2 дозволяє зробити висновки про те, що у них є багато моментів, які можуть впливати на достовірність результатів моделювання та зменшувати ефективність автоматизованої системи обробки даних СВНО.

До числа таких можна віднести:

1. Припущення про доцільність застосування лінійної апроксимації при роботі з окремими сегментами між точками множини  $M_i = \{(x_1, y_1), \dots, (x_{k_i}, y_{k_i})\}$  при формуванні неперервного апроксимаційного маршруту чи  $M_1$ , чи  $M_2$ , є сумнівним і нічим не підтвердженим. Особливий сумнів виникає у випадку значної віддаленості в часі або просторі досліджуваних точок. А тому актуальності набуває завдання дослідження достовірності результатів на заданих експериментальних наборах даних у разі, якщо застосовуватиметься інший вид апроксимації (в тому числі за допомогою сплайнів). При цьому, окремої уваги потребує питання відповідності обраного методу апроксимації логічному змісту процесу руху суден.

2. Дослідження маршрутів різних суден на предмет формування кластерів може здійснюватись не лише в межах між пунктами відправлення та призначення, а й між довільними двома точками, в межах яких існує особлива необхідність вивчення питання про порушення прикордонного законодавства. У цьому випадку застосування формули (2) для відшукування метрики  $R(M_1, M_2)$ , формули (6) для відшукування метрики  $R(M_2, M_1)$ , або формули (7) для встановлення метрики  $R^k(M_1, M_2)$ , потребує обґрунтування вибору кількостей  $k_2$  і  $k_1$  необхідних для дослідження точок, які б забезпечили достатній рівень достовірності висновків, що формуються.

3. Для умов, що наведені вище, актуальним є питання і вибору для дослідження конкретних точок на маршрутах  $M_1$  та  $M_2$ , які при реалізації значень  $k_2$  і  $k_1$  забезпечували б достатній рівень достовірності висновків, що формуються. Актуальність дослідження цього питання обумовлюється тим, що при реалізації різних комбінацій досліджуваних точок величина  $R(M_1, M_2)$ , або  $R(M_2, M_1)$ , або  $R^k(M_1, M_2)$  може суттєво відрізнятися, що може впливати на достовірність методики кластеризації.

4. Виникає сумнів і питання забезпечення об'єктивності результатів, що визначаються застосуванням метрики (7). Адже властивість комутативності метрики справедлива не лише у випадку застосування (7). Так, при застосуванні метрики:

$$R^k(M_1, M_2) = \max(R(M_1, M_2), R(M_2, M_1)), \quad (9)$$

або

$$R^k(M_1, M_2) = \frac{R(M_1, M_2) + R(M_2, M_1)}{2}, \quad (10)$$

або

$$R^k(M_1, M_2) = \sqrt{R(M_1, M_2)R(M_2, M_1)}, \quad (11)$$

існує велика ймовірність її комутативності, а отже, можливості застосування. Однак висновки щодо подібності маршрутів та співпадання меж кластерів при застосуванні наведених метрик у різних випадках можуть відрізнятися. А отже, питання адекватності застосовуваного інструментарію та достовірності результатів, що ним забезпечуються, потребують окремого вивчення.

5. При застосуванні метрик (2), або (6), або (7) може мати місце випадок, коли для окремих сегментів між точками  $(x_i, y_i)$  і  $(x_{i+1}, y_{i+1})$  множини  $M_1$  або  $M_2$  виконуватиметься умова  $x_i = x_{i+1}$ . Тобто у вибраній системі координат окремі сегменти можуть бути паралельними осі ординат. У цьому випадку формула (5) не може бути застосовною для відшукування величини  $R_{min j}$ .

6. У разі, якщо деякий досліджуваний маршрут  $M_i$  на окремій ділянці маршруту між пунктом відправлення та призначення паралельний еталонному маршруту  $M_{et}$ , але на цій ділянці величина

$R(M_i, M_{st}) > r_{pk}$ , то це може спричинити не включення маршруту  $M_i$  у кластер, хоча у загальному випадку цей маршрут може увійти до цього кластеру. Разом з тим, існуюча методика кластеризації цієї особливості не враховує. А оскільки така особливість важлива з позиції формування висновків щодо наявності порушень прикордонного законодавства, то відповідне питання також потребує окремого дослідження.

7. У разі достатньої складності маршруту існуюча методика може бути незастосовною, оскільки може не відповідати логічному змісту досліджуваного процесу.

8. Формування еталонного маршруту  $M_{st}$  описане у роботі [5]. Цей маршрут являє собою набір дискретних точок. Актуальним є питання формування такого методу побудови еталонного маршруту  $M_{st}$ , який би забезпечував створення еталонного маршруту у вигляді неперервної траєкторії.

**Висновки.** Проведене дослідження дозволяє зробити висновок про те, що базовим положенням, яке дійсно визначає ефективність автоматизованої системи обробки даних, є метод кластеризації маршруту суден. Існуючий метод кластеризації характеризується рядом особливостей, які наведені вище та стосуються інструментальних особливостей реалізації і впливу на достовірність результатів для подальшої оцінки ризиків протиправної діяльності на державному кордоні. Аналіз визначених особливостей дозволяє зробити висновок про те, що існуючий метод кластеризації маршрутів суден не може у явному вигляді бути застосовним для використання в СВНО як процедурний модуль автоматизованої системи обробки даних, який би забезпечував достатній рівень достовірності виявлення ознак порушення прикордонного законодавства.

Напрямами подальших досліджень вбачається вивчення питань, які окреслені проблемними моментами 1-8 застосування існуючого методу кластеризації, а також їх подальша формалізація, алгоритмізація, програмування та апробація.

### Література

1. Боровик О. В., Рачок Р. В., Мазур В. Ю. Визначення підходів до просторового аналізу даних в інформаційно-телекомунікаційній системі морської охорони «Гарт-12» // Збірник наукових праць № 1 (75). Серія: Військова та технічні науки. – Хмельницький: Вид. НАДПСУ, 2018. – С. 134-143.
2. Мазур В. Ю., Боровик О. В. Методичні основи оцінки ефективності функціонування єдиної системи висвітлення надводної обстановки на морській ділянці. – Наука і техніка Повітряних Сил Збройних Сил України № 2 (31). – Харків: ХНУПС, 2018. – С. 182-189.
3. Мазур В. Ю., Боровик О. В., Рачок Р. В. Метод кластеризації маршрутів суден в системі висвітлення надводної обстановки // Сучасні інформаційні технології у сфері безпеки та оборони Вип. № 2(32). – К.: Національний університет оборони України імені Івана Черняховського, Інститут інформаційних технологій, 2018. – С. 87-92.
4. Pan Sheng, Jingbo Yin Extracting Shipping Route Patterns by Trajectory Clustering Model Based on Automatic Identification System Data. - Sustainability 2018, 10.
5. Боровик О. В., Рачок Р. В., Мазур В. Ю. Методика виявлення просторових аномалій руху суден та її використання при оцінюванні ризиків у системі морської охорони «Гарт-12» // Наука і оборона. – 2018. – № 2. С. 65-69.
6. Arguedas V. F., Pallotta G., Vespe M. Maritime Traffic Networks: From Historical Positioning Data to Unsupervised Maritime Traffic Monitoring. - Transactions on intelligent transportation systems, Vol. 19, № 3, March 2018. - P. 722-732.

### References

1. Borovyk O.V., Rachok R.V., Mazur V.Yu. Definition of approaches to spatial data analysis in the information and telecommunication system of maritime protection "Gart-12" // Collection of scientific works № 1 (75). Series: Military and technical sciences. - Khmelnytsky: Ed. NADPSU, 2018. - P. 134-143.
2. Mazur V. Yu., Borovyk O.V. Methodical bases of an estimation of efficiency of functioning of uniform system of illumination of a surface situation on a sea site. - Science and Technology of the Air Force of the Armed Forces of Ukraine № 2 (31). - Kharkiv: KhNUPS, 2018. - P. 182-189.
3. Mazur V. Yu., Borovyk O.V., Rachok R.V. The method of clustering of ship routes in the system of surface lighting // Modern information technologies in the field of security and defense Vol. № 2 (32). - Kyiv: Ivan Chernyakhovsky National University of Defense of Ukraine, Institute of Information Technologies, 2018. - P. 87-92.
4. Pan Sheng, Jingbo Yin Extracting Shipping Route Patterns by Trajectory Clustering Model Based on Automatic Identification System Data. - Sustainability 2018, 10.
5. Borovyk O.V., Rachok R.V., Mazur V.Yu. Methods for detecting spatial anomalies in the movement of ships and its use in risk assessment in the marine security system "Gart-12" // Science and Defense. - 2018. - № 2. - S. 65-69.
6. Arguedas V. F., Pallotta G., Vespe M. Maritime Traffic Networks: From Historical Positioning Data to Unsupervised Maritime Traffic Monitoring. - Transactions on intelligent transportation systems, Vol. 19, № 3, March 2018. - P. 722-732.

Надійшла / Paper received: 22.08.2020  
Надрукована / Paper Printed : 03.11.2020

## Додаток В

### Презентація по темі

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
Кафедра комп'ютерної інженерії та системного  
програмування

# Кіберфізична система виявлення просторових аномалій руху суден на основі методу кластеризації маршрутів

ВИКОНАЛА СТ. ГРУПИ КІ2М-19-1:  
КОСТЕЛЬНА Т.В.

НАУКОВИЙ КЕРІВНИК:  
Д.Т.Н., ПРОФ. БОРОВИК О.В.

2

Метою дипломної роботи є розробка кіберфізичної системи виявлення просторових аномалій руху суден на основі покращеного методу кластеризації маршрутів.

#### Задачі дослідження:

- Розглянути особливості та методи на основі яких базуються сучасні системи висвітлення надводної обстановки ;
- Провести аналіз існуючих засобів та підходів до виявлення просторових аномалій руху суден;
- Ознайомитись з методом кластеризації, а також із особливостями його використання в системах виявлення просторових аномалій руху суден;
- Розробити та реалізувати кіберфізичну систему виявлення просторових аномалій руху суден на основі покращеного методу кластеризації маршрутів руху суден.

## Аналіз предметної області

### Переваги кіберфізичних систем:

- Взаємодія між рівнями системи забезпечується самою системою і не вимагає від проєктувальників різних рівнів кіберфізичної системи організувати взаємодію з іншими рівнями;
- У разі модифікації функціональності на будь-якому з рівнів кіберфізичної системи, не виникає потреби в модифікації інших рівнів;
- Користувач повинен знати особливості використання системи тільки на своєму рівні, що дозволяє пришвидшувати впровадження кіберфізичної системи за рахунок спрощення навчання користувачів;
- Надзвичайні обчислювальні потужності та великий об'єм сховища інформації;
- Різноманітні способи отримання та виведення різноманітної інформації;
- Жорсткі вимоги до надійності, захищеності від несанкціонованого проникнення та використання, малопомітності.

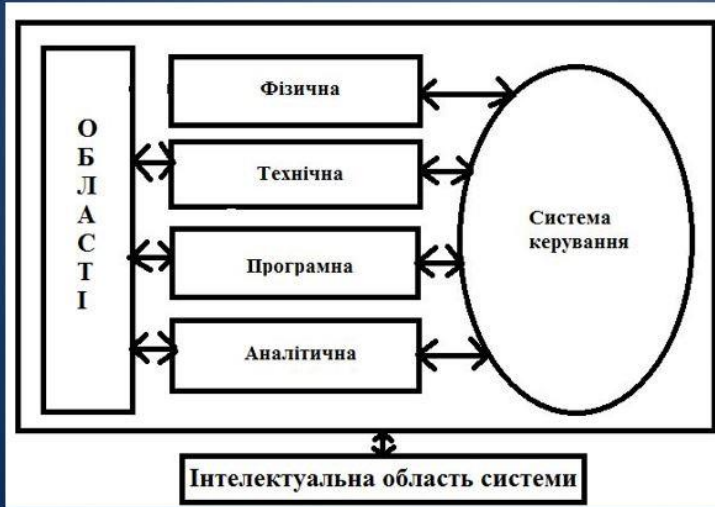
## Постановка задачі

Важливим чинником стабільності держави є надійна охорона кордону. В сучасних умовах постійного зростання обсягів даних, які необхідно враховувати при прийнятті рішень, надзвичайно важливою є інформаційна складова сучасної моделі охорони кордону. Ця складова реалізується з використанням інтегрованої інформаційно-телекомунікаційної системи «Гарт-12». Система «Гарт-12» надає можливість отримання інформації про поточне положення суден, порт відправки та порт призначення, тип вантажу та інші додаткові дані. При виборі судна у системі висвітлення надводної обстановки існує можливість перегляду попередніх точок його маршруту, отриманих в певні дискретні моменти часу.

По даній інформації, навіть експертам може бути важко визначити потенційні загрози, що стосуються порушення прикордонного законодавства. Оскільки ж відхилення судна від еталонного маршруту в окремих випадках може являти собою порушення правил прикордонного режиму, то актуальним завданням є ідентифікація тих ознак, які можуть вказувати на зазначене порушення.

Виявлення цих ознак є достатньо складним завданням, оскільки пов'язане з необхідністю обробки значних обсягів даних щодо надводної обстановки в ІТС «Гарт-12» та складністю аналізу ризиків. Для вирішення цього завдання запропоновано підхід, що стосується створення відповідної автоматизованої системи обробки даних, яка б містила проєдну кластеризації маршрутів, встановлення еталонних маршрутів руху суден в СВНО, виявлення аномалій руху суден і оцінки ризиків

# Загальна схема кіберфізичної системи



Головна ідея кіберфізичних систем тісно пов'язана з роботичними та сенсорними системами, які обладнані «розумними» механізмами з достатніми обчислювальними можливостями для достатнього керування.

# Система висвітлення надводної та підводної обстановки

## Метод кластеризації

Метод обробки даних, що полягає у встановленні в певній сукупності за певним алгоритмом членів, які є подібними.

Для проведення кластеризації маршрутів потрібно на основі введених даних побудувати матрицю.

Після формування матриці, для кожного її рядка обчислюється:

- кількість елементів, які не перевищують порогового значення для кластеризації ;
- величина, яка формується лише з тих значень, для яких виконується умова порогового значення.

На основі отриманих даних, які сортуються за зменшенням (для елементів з однаковими значеннями здійснюється сортування за збільшенням), формується впорядкована множина.

Далі множина використовується для визначення тих маршрутів, які можуть бути включені до одного і того ж кластера.

8

## Метод кластеризації

### Алгоритм формування кластеру:

- ▶ якщо для першого елемента множини  $M_s$  виконується умова  $n_s(i) > 0$ , то створюється кластер з центральним маршрутом  $i$  в нього включаються всі маршрути  $j$ , для яких  $R^k(M_i, M_j) \leq \tau_{pk}$ ;
- ▶ з множини  $M_s$  видаляються всі набори  $(i, n_s(i), s_v(i))$ , в яких перший елемент  $i$  співпадає із значеннями, які визначили маршрути, що увійшли до створеного кластера;
- ▶ кроки 1-2 алгоритму повторюються до тих пір, поки потужність множини  $M_s$  є більшою за 0.

## Нелінійна інтерполяція

Для табулювання швидкозмінних функцій потрібний дуже малий крок, тобто виникає необхідність створювати таблиці дуже великих обсягів, що в деяких випадках неможливо. Виявляється, що перетворенням змінних можна домогтися того, щоб в нових змінних графік був близький до прямої хоча б на окремих ділянках. В цьому випадку інтерполяцію проводять в змінних, а потім зворотньою інтерполяцією знаходять  $y_i = y(\eta_i)$ .

Перетворення повинні бути досить простими. При цьому треба дбати про те, щоб і зворотне перетворення було нескладним.

Отримати формулу для нелінійної двоточкової інтерполяції функції  $y(x)$  можна, якщо змінні можна перетворити за формулою  $\eta = \ln(y)$ .

Складемо інтерполяційний поліном Ньютона на двоточковому шаблоні:  $\eta = \eta_0 + \frac{\eta_1 - \eta_0}{\xi_1 - \xi_0} (\xi - \xi_0)$

У вихідних змінних маємо  $\ln(y) = \ln(y_0) + \frac{\ln(y_1) - \ln(y_0)}{x_1 - x_0} (x - x_0)$

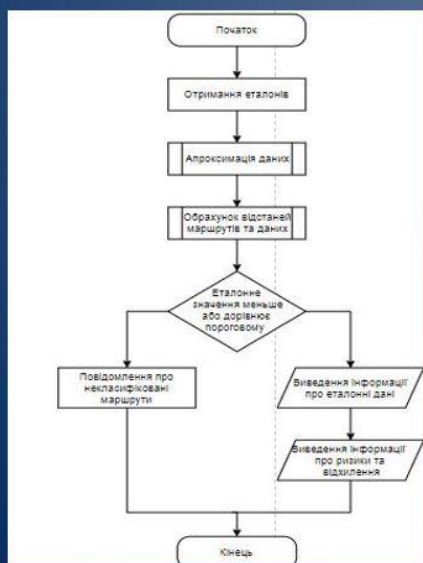
і остаточно  $y = y_0 (y_1 / y_0)^{(x - x_0) / (x_1 - x_0)}$

## Загальний алгоритм роботи системи на основі методу кластеризації



- Відбувається введення в систему базових даних про маршрути;
- Підготовка даних та їх наступна апроксимація;
- Розрахунок даних про маршрути;
- Представлення користувачеві результатів обчислень.

## Алгоритм побудови еталонного маршруту



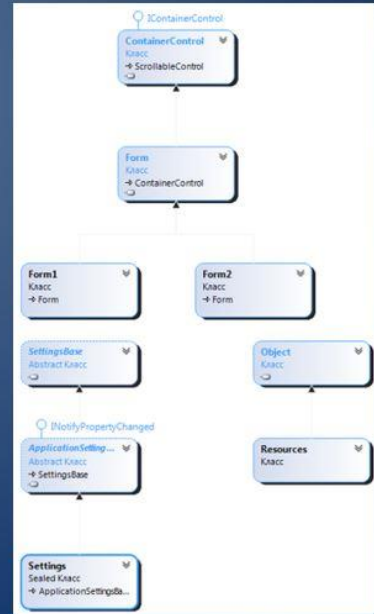
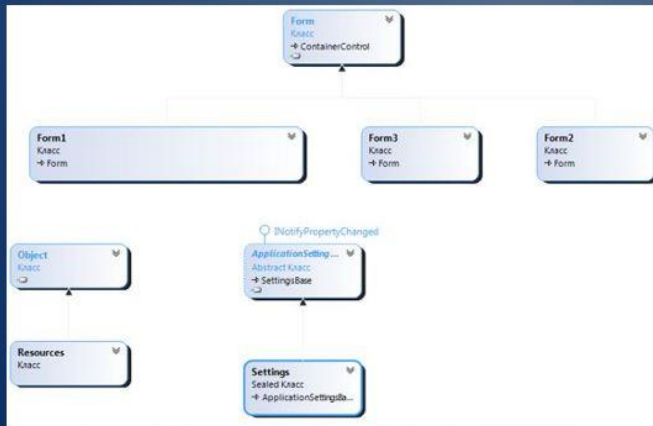
- Отримання еталонів;
- Апроксимація даних;
- Відбувається обрахування відстаней маршрутів та даних;
- Аналізується чи еталонне значення менше або дорівнює пороговому;
- В залежності від еталонного значення виводиться інформація про некластичності маршруту або про ризики відхилення.

## Алгоритм виявлення просторових аномалій руху суден

- Встановлення подібності маршрутів довільних пар різних суден;
- Створення групи маршрутів між визначеними пунктами відправлення та призначення (транспортний коридор між двома пунктами);
- Визначення еталонного маршруту між двома точками в межах сформованого кластера;
- Визначення належності траєкторії руху судна кластеру маршрутів між даними пунктами відправлення та призначення;
- Оцінка ступеня відхилення траєкторії досліджуваного судна від еталонного маршруту.

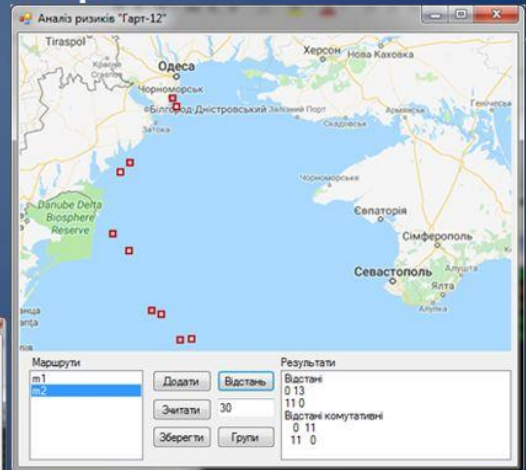
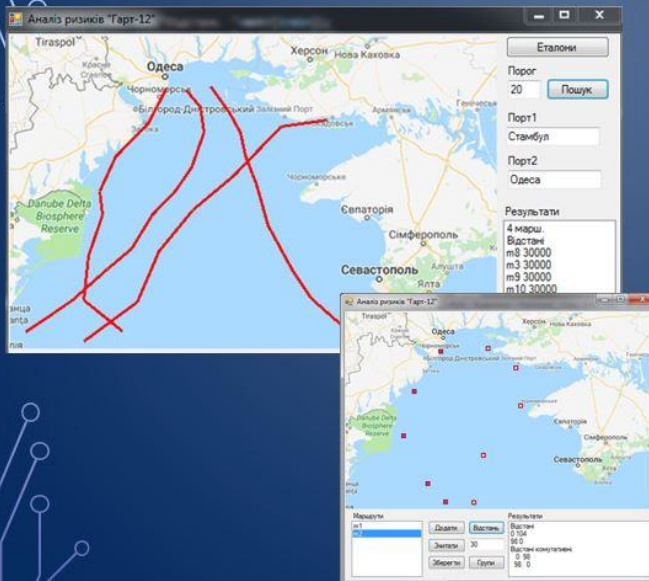
13

## Діаграми класів системи висвітлення надводної обстановки



14

## Програмна реалізація



## Висновки

- ▶ В результаті виконаного наукового дослідження розроблена кіберфізична система виявлення просторових аномалій руху суден на основі удосконаленого методу кластеризації даних;
- ▶ Удосконалено метод кластеризації маршрутів руху суден як процедурного модуля автоматизованої системи обробки даних у системі висвітлення надводної обстановки;
- ▶ Набула подальшого розвитку інформаційна технологія кластеризації маршрутів руху суден у системі висвітлення надводної обстановки.
- ▶ За темою дипломної роботи була опублікована одна стаття у Міжнародному науковому журналі «Комп'ютерні системи та інформаційні технології».

Ім'я користувача:  
Кафедра КІ

Дата перевірки:

25.05.2021 11:09:05 EEST

Дата звіту:

25.05.2021 11:37:09 EEST

ID перевірки:

1008004346

Тип перевірки:

Doc vs Internet + Library

ID користувача:

100005591

Назва документа: Кіберфізична система виявлення просторових аномалій руху суден на основі методу клас... Кількість сторінок: 72 Кількість слів: 11347 Кількість символів: 85987 Розмір файлу: 3.89 MB ID файлу: 1008095688

## 23.2%

### Схожість

Найбільша схожість: 13.1% з Інтернет-джерелом (<http://elar.khnu.km.ua/jspui/bitstream/123456789/9960/1/23-%D0%A2>)

21.9% Джерела з Інтернету

146

Сторінка 74

1.67% Джерела з Бібліотеки

64

Сторінка 75

### 0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

## 0%

### Вилучень

Немає вилучених джерел

### Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіт

Замінені символи

16



Ім'я користувача:  
Кафедра КІ

ID перевірки:  
1008067205

Дата перевірки:  
28.05.2021 14:41:30 EEST

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
28.05.2021 14:51:15 EEST

ID користувача:  
100005591

Назва документа: Кіберфізична система виявлення просторових аномалій руху суден на основі методу клас...

Кількість сторінок: 137 Кількість слів: 17970 Кількість символів: 130065 Розмір файлу: 6.31 MB ID файлу: 1008153827

## 13.6% Схожість

Найбільша схожість: 1.61% з джерелом з Бібліотеки (ID файлу: 1007657363)

12.2% Джерела з Інтернету 563 ..... Сторінка 139

2.22% Джерела з Бібліотеки 100 ..... Сторінка 144

## 0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

## 60.1% Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 8 слів та 0%)

7.82% Вилучення з Інтернету 1 ..... Сторінка 145

59.7% Вилученого тексту з Бібліотеки 1 ..... Сторінка 145

## Модифікації

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

Замінені символи 18

## Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 1.0%

Словари проверки: en\_US, ru\_RU, ua\_UA. Ошибок в документах: 6%

ID: 91228 Название: Кіберфізична система виявлення просторових аномалій руху суден на основі методу кластеризації маршрутів Добавлено в БД: 2021-05-25 Авторы: Костельна Т.В. Руководители: Боровик О.В. Консультанты: Оponentы:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	73616	613	2530 (3%)	34 (6%)

### Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

Дипломник: Костельна Т.В.

Тема: Кіберфізична система виявлення просторових аномалій руху суден на основі методу кластеризації маршрутів

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг дипломної роботи:

Кількість сторінок записки 61

1. Короткий зміст роботи та прийнятих рішень

У магістерській роботі здійснено аналіз існуючих методів і засобів виявлення просторових аномалій руху суден; обґрунтовано можливість застосування методу кластеризації або визначення шляхів його удосконалення для використання в системі висвітлення надводної обстановки; розроблена кіберфізична система виявлення просторових аномалій руху судна на основі удосконаленого методу кластеризації даних про маршрути руху суден.

2. Висновок про відповідність роботи дипломному завданню

Магістерська робота виконана у відповідності до дипломного завдання.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи:

У першому розділі проведено аналіз існуючих методів і засобів виявлення просторових аномалій руху суден. У результаті проведеного дослідження встановлено, що існуючий метод не може у явному вигляді бути застосовним для використання в системі висвітлення надводної обстановки як процедурний модуль автоматизованої системи обробки даних, який би забезпечував достатній рівень достовірності виявлення ознак порушення прикордонного законодавства.

У другому розділі автор розглянув новітні підходи до оцінки відхилення маршрутів суден. Також автором був більш детально проаналізований і в результаті удосконалений існуючий метод кластеризації маршрутів суден.



Завідувачу кафедри КІСП

д-р.техн.наук, проф. Говорущенко Т. О.

Костельна Тамара Валентинівна

---

ПІБ здобувача вищої освіти

ФПКТС, 2 курсу, групи КІ2М-19-1

### ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіатоповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

25.05.2021

дата



підпис

## РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА СИСТЕМНОГО ПРОГРАМУВАННЯ

### ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

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

Назва: «Кіберфізична система виявлення просторових аномалій руху суден на основі методу кластеризації маршрутів»

Автор: Костельна Тамара Валентинівна

Спеціальність: 123 – Комп'ютерна інженерія та програмування

Освітня програма: освітньо-наукова

Науковий керівник: Боровик Олег Васильович, д.т.н, професор

Після аналізу звіту подібності зроблено такий висновок:

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

Підтвердження:

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

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 10.2% і адресується до 70 першоджерел, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи



О.В. Боровик

Гарант ОП



О. С. Савенко

Завідувач кафедри КІСП



Т. О. Говорущенко