

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ
імені адмірала Макарова

Навчально-науковий інститут автоматики і електротехніки

(повне найменування інституту, назва факультету)

Кафедра програмованої електроніки, електротехніки і телекомунікацій

(повна назва кафедри)

«Допущений до захисту»

Завідувач кафедри



В. М. Рябенський

д.т.н., професор

« 16 » грудня 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «магістр»

на тему:

Дослідження впливу погодних умов на роботу АІС

Виконав: студент 6 курсу, групи 6321М

спеціальності 171

«Електроніка»

(шифр і назва спеціальності)

ОПП

«Електронні системи»

Вишневський В. В.



(прізвище та ініціали студента)

Керівник

Дьяконов О. С.



(прізвище та ініціали)

Рецензент

Іхсанов Ш. М.



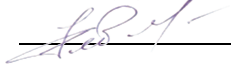
(прізвище та ініціали)

м. Миколаїв – 2024 року

Національний університет кораблебудування імені адмірала Макарова

ННІ	автоматики та електротехніки
Кафедра	програмованої електроніки, електротехніки і телекомунікацій
Рівень вищої освіти	другий (магістерський)
Спеціальність	171 «Електроніка»
ОПП	Електронні системи

ЗАТВЕРДЖУЮ

Завідувач кафедри  В. М. Рябенський
д.т.н., професор

« 16 » жовтня 2024 р.

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ АТЕСТАЦІЙНУ РОБОТУ СТУДЕНТУ

Вишневському Владиславу Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження впливу погодних умов на роботу АІС

керівник роботи Дьяконов Олексій Сергійович, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу
від «14» жовтня 2024 року № 1075-уч.

2. Строк подання студентом роботи грудень 2024 р.

3. Вихідні дані до роботи колекція даних з сайту MarineTraffic, dxinfocentre.

4. Мета дослідження .

Дослідити вплив аномального поширення радіохвиль на роботу системи АІС.

Створити нейромережу для прогнозування завантаженості АІС станції на основі карт тропосферного індексу

5. Зміст розрахунково-пояснювальної записки
(перелік питань, які потрібно розробити)

1. Аналіз механізмів розповсюдження УКХ радіохвиль

2. Дослідження впливу тропосферних явищ на роботу АІС

3. Розробка нейромережі для прогнозування площі покриття АІС станції

4. Охорона праці

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
презентація у PowerPoint, Мапа Нерburn Тропо Index (НТІ), Вихідні дані,

Інструменти для скрапінгу, Схема підготовки даних для навчання нейромережі

Приклад підготовлених даних, Схема навчання мережі, Приклад прогнозу моделі

Діаграма ступеню збігу маски покриття між даними з marinetraffic та прогнозом
мережі, Графік залежності кількості сигналів від площі

7. Консультанти розділів роботи

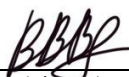
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Тубальцев А.М.		

8. Дата видачі завдання « 16 » жовтня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1.	Системи автоматичної ідентифікації суден (АІС): принципи роботи та розповсюдження радіосигналу	01.10.2024-25.10.2024	Вик.
2.	Аномальне поширення УКХ-радіохвиль. Вплив на роботу АІС	25.10.2024-10.11.2024	Вик.
3.	Дослідження можливості прогнозування завантаження АІС за погодними картами розповсюдження УКХ	10.11.2024-27.11.2024	Вик.
4.	Охорона праці	27.11.2024-30.11.2024	Вик.

Студент


(підпис)

В. В. Вишневецький

(прізвище та ініціали)

Керівник роботи


(підпис)

О. С. Дьяконов

(прізвище та ініціали)

Анотація

У дипломній роботі проведено аналіз автоматичної ідентифікаційної системи (AIC) з урахуванням впливу тропосферних явищ на розповсюдження УКХ. Особливу увагу приділено дослідженню аномальних явищ у тропосфері, які формують хвилеводи, що значно розширюють зону дії AIC. У рамках дослідження було розроблено нейронну мережу, яка на основі карт тропосферного розповсюдження прогнозує зону покриття станції. Для уточнення взаємозв'язку між зоною покриття станції та кількістю прийнятих сигналів створено модель лінійної регресії. Завдяки цій моделі було визначено кількість сигналів, спрогнозованих розробленою нейромережею. Отримані результати демонструють можливості підвищення ефективності функціонування AIC шляхом урахування тропосферних аномалій і використання сучасних методів машинного навчання для моделювання зон покриття та прогнозування параметрів роботи станцій.

Ключові слова: УКХ-хвилі, , тропосферні аномалії, нейронна мережа, прогнозування сигналів.

Abstract

This thesis conducts an analysis of automatic identification system (AIS), focusing on the effects of tropospheric phenomena on the propagation of UHF radiowaves. The study delves into anomalous tropospheric events that create waveguides, substantially extending the AIS coverage range. A neural network was developed to forecast the station coverage area based on tropospheric propagation maps. To quantify the correlation between the station coverage area and the number of received signals, a linear regression model was constructed. This model was employed to estimate the number of signals predicted by the neural network. The findings indicate the potential to enhance AIS operational efficiency by considering tropospheric anomalies and leveraging advanced machine learning techniques for modeling coverage areas and forecasting station performance parameters.

Keywords: VHF radiowaves, , tropospheric anomalies, signal prediction, neural network

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень та термінів.....	7
Вступ.....	9
Розділ 1. Система автоматичної ідентифікації суден (АІС): принципи роботи та розповсюдження радіосигналу.....	11
1.1. Автоматична ідентифікаційна система (АІС)	11
1.2. УКХ канали АІС	16
1.3. Особливості розповсюдження радіохвиль УКХ-діапазону	19
1.4. Типові градієнти атмосферної рефракції та поширення радіохвиль у діапазоні УКХ	24
Розділ 2. Аномальне поширення укх радіохвиль. Вплив на роботу АІС ...	31
2.1. Аномальні профілі градієнтів рефрактивності для утворення хвилеводів радіосигналів у діапазоні УКХ	31
2.2. Деякі проблеми, пов'язані з тропосферними хвилеводами	37
2.3. Приклади впливу аномального поширення УКХ-радіохвиль на роботу АІС.....	38
2.4. Можливість прогнозування завантаженості VDL каналу АІС станцій на основі тропосферних карт.....	46
Розділ 3. Дослідження можливості прогнозування завантаження АІС за погодними картами тропосферного індексу	50
3.1. Вихідні дані та постановка задачі.....	50
3.2. Обробка та підготовка даних	54
3.3. Вибір моделі. Навчання нейромережі	66
3.4. Залежність кількості прийнятих АІС-повідомлень від площі зони покриття.....	78
Розділ 4. Охорона праці	83
4.1. Аналіз шкідливих та небезпечних факторів на робочих місцях	83
4.2. Вимоги до організації робочого місця	84
4.3. Заходи для запобігання шкідливим факторам.....	87
4.4. Домедична допомога постраждалим від електричного струму	89
Висновки	91

Список використаних джерел	92
Додаток А Код програм для скрапінгу, та збору даних	94
Додаток Б Код програми Create_geotiff	98
Додаток В Код програми getTropoData.....	100
Додаток Г Код програми для створення мапи тропосферного індексу.....	102
Додаток Ґ Код програми Station_Area_creator.py	104
Додаток Д Код програми для створення бінарних масок	107
Додаток Е Код для тренування нейромережевої моделі.....	108
Додаток Є Код для оцінки прогнозів нейромережі	111
Додаток Ж Код для побудови моделі лінійної регресії.....	113

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

API	– Application Programming Interface;
CNN	– Convolutional neural network;
CPU	– central processing unit;
CSS	– Cascading Style Sheets;
CSTDMA	– Carrier Sense Time-Division Multiple Access;
CSV	– Comma-Separated Values;
DSC	– Digital Signal Controler;
ETA	– Estimated time of arrival;
GMDSS	– Global Maritime Distress and Safety System;
GNSS	– Global Navigation Satellite System;
GPS	– Global Positioning System;
GPU	– graphics processing unit;
GSM	– Groupe Spécial Mobile;
HSV	– Hue, Saturation, Value;
HTI	– Hepburn Tropo Index;
HTML	– HyperText Markup Language;
IMO	– International Maritime Organization;
JSON	– JavaScript Object Notation;
LOS	– Line of sight;
MMSI	– Maritime Mobile Service Identity;
PNG	– portable network graphics;
RGB	– Red, Green, Blue;
ROT	– Rate of Turn;
SNR	– signal-to-noise ratio;
SOLAS	– International Convention for the Safety of Life at Sea;
SOTDMA	– Self-organized time-division multiple access;
SVG	– Scalable Vector Graphics;

TDMA	– Time Division Multiple Access;
TPU	– Tensor Processing Unit;
UHF	– Ultra high frequency;
VDL	– VHF datalink loading;
VHF	– Very High Frequency;
XHR	– XMLHttpRequest;
AIC	– Автоматична ідентифікаційна система;
УКХ	– ультракороткі хвилі.

					171.6321М.КР.ПЗ.Скорочення	Арк.
						8
Змн.	Арк.	№ документу	Підпис	Дата		

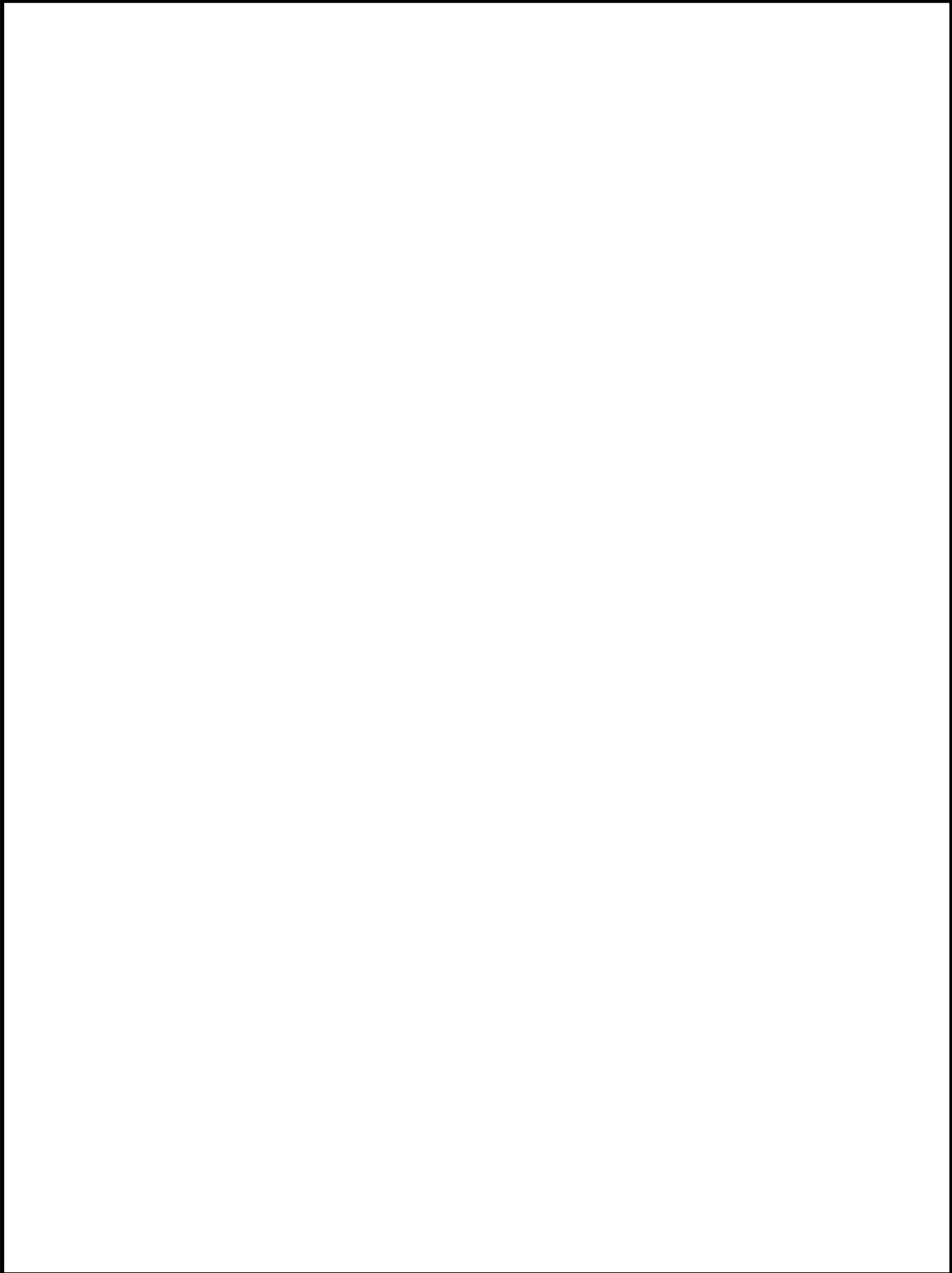
ВСТУП

Автоматична ідентифікаційна система (АІС) – один з ключових елементів сучасних засобів морської навігації та безпеки. Вона використовує УКХ-діапазон хвиль. Ефективність роботи берегових станцій значною мірою залежить від умов розповсюдження радіохвиль, які можуть змінюватися через аномальні явища у тропосфері.

Тропосферні аномалії, такі як утворення хвилеводів, здатні істотно впливати на зону дії АІС, розширюючи її за межі номінального діапазону (20 морських міль). Це може значно вплинути на завантаженість VDL-каналу АІС, що може бути критично для роботи системи.

У роботі розглянуто тропосферні процеси, що впливають на розповсюдження радіохвиль, та проаналізовано їхній вплив на функціонування АІС. Для прогнозування зон покриття розроблено нейронну мережу, яка базується на картографічних даних тропосферного індексу. Крім того, створено модель лінійної регресії, яка дозволяє дослідити взаємозв'язок між зоною покриття станції та кількістю прийнятих сигналів.

					171.6321М.КР.ПЗ.Вступ	Арк.
						9
Змн.	Арк.	№ документу	Підпис	Дата		



					171.6321М.КР.ПЗ.Р1			
Змн.	Арк.	№ документу	Підпис	Дата	Система автоматичної ідентифікації суден (АІС): принципи роботи та розповсюдження радіосигналу	Літ.	Арк.	Аркуші
Розробник		Вишневський В. В.						
Консультант							10	
Н. контр.		Добрінова Л. П.				НУК імені адмірала Макарова		
Керівник		Дьяконов О. С.						
Зав. каф.		Рябенський В. М.						

РОЗДІЛ 1. СИСТЕМА АВТОМАТИЧНОЇ ІДЕНТИФІКАЦІЇ СУДЕН (АІС): ПРИНЦИПИ РОБОТИ ТА РОЗПОВСЮДЖЕННЯ РАДІОСИГНАЛУ

1.1. Автоматична ідентифікаційна система (АІС)

Автоматична ідентифікаційна система (АІС) [1] – це сучасна система радіозв'язку, що використовується в морській навігації для автоматичного обміну інформацією між суднами та береговими станціями (рис 1.1). Основна мета АІС – підвищення безпеки мореплавання шляхом забезпечення своєчасного обміну точними даними про місцезнаходження, курс, швидкість та інші параметри рухомих об'єктів.

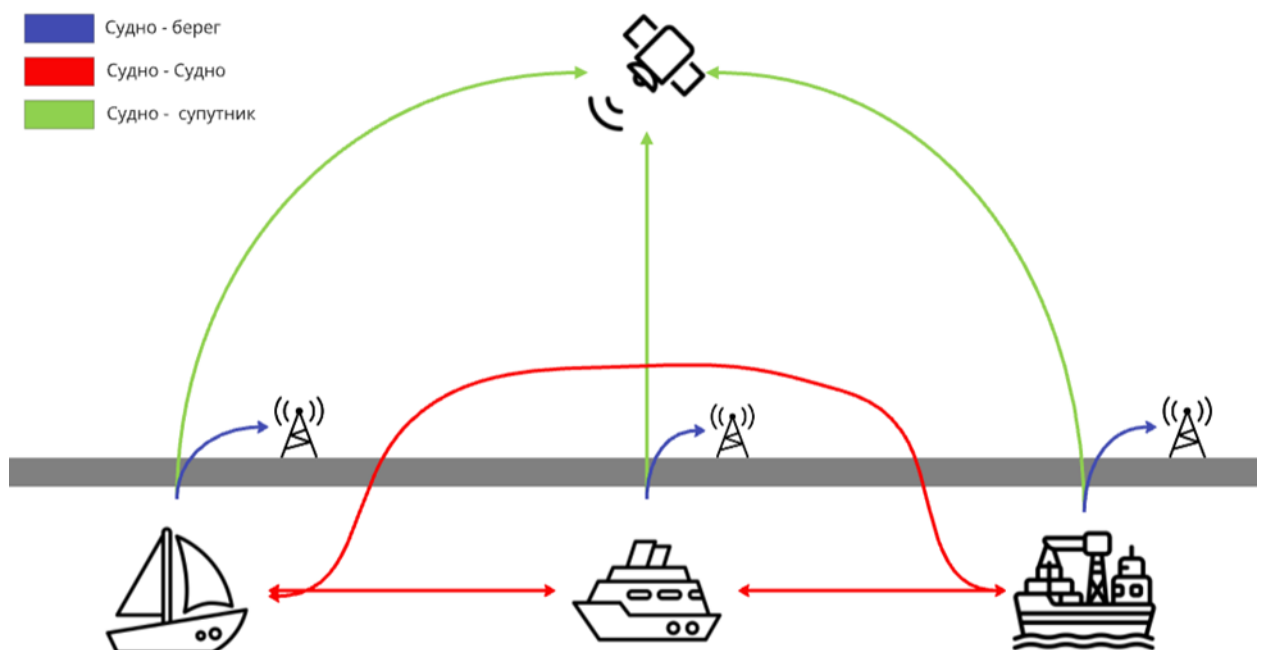


Рисунок 1.1 – Схема роботи АІС

АІС працює в діапазоні ультракоротких хвиль (УКХ), використовуючи діапазон 156,025 – 162,025 МГц. Інформація передається у вигляді цифрових повідомлень, які містять як динамічні дані (курс, місцезнаходження, швидкість, отримані від навігаційних датчиків), так і статичні дані (порт приписки, назва судна, осадка, розміри, тип судна, номер MMSI та ін.).

Кожне судно, що оснащено VHF-радіостанцією з функцією цифрового селективного зв'язку (DSC) повинно мати унікальний ідентифікаційний

номер MMSI [2]. Ця вимога поширюється на більшість суден довжиною понад 20 метрів, а також на деякі менші судна залежно від національних правил. Інші морські об'єкти, такі як бурові платформи та морські літаки, також можуть мати MMSI.

MMSI є невід'ємною частиною глобальної морської системи зв'язку для забезпечення безпеки (GMDSS). Ця система, впроваджена Міжнародною морською організацією (IMO), стандартизує радіозв'язок між судами та між судами і береговими станціями.

MMSI використовується для:

- Унікальної ідентифікації: Кожен MMSI надається лише одному судну, що дозволяє точно ідентифікувати його в будь-якій точці світу.
- Безпеці: MMSI використовується для передачі сигналів лиха, повідомлень про місцезнаходження та інших важливих даних. Це допомагає швидко реагувати на надзвичайні ситуації.
- Відстеження: Система AIS використовує MMSI для відстеження суден у реальному часі, що допомагає запобігати зіткненням.
- Ефективного управління: Морська влада використовує MMSI для контролю руху суден та забезпечення безпеки на водних шляхах.

Мобільні станції AIS працюють в автономному режимі, періодично передаючи інформацію про судно. Крім того, вони можуть відповідати на запити інших станцій або за потреби передавати додаткову інформацію.

Інформація AIS, котра приймається та передається

Інформація поділяється на статичну, динамічну, та рейсову:

Статична інформація

Ці дані вводяться один раз при встановленні системи AIS та залишаються незмінними:

- MMSI (Ідентифікаційний номер морської рухомої служби).
- Ідентифікаційний номер IMO (при наявності).
- Назва судна.

					171.6321M.KP.ПЗ.Р1	Арк.
Змн.	Арк.	№ документа	Підпис	Дата		12

- Тип судна.
- Габарити судна .
- Позиція антен GNSS (відносно судна).

Динамічна інформація

Ця інформація змінюється в реальному часі та автоматично передається:

- Поточні координати (широта та довгота).
- Курс відносно ґрунту.
- Швидкість відносно ґрунту.
- Курс повороту (Rate of Turn, ROT).
- Географічний напрямок (Heading).
- Час останнього визначення координат (синхронізується за GPS).

Рейсова інформація

Ці дані вводяться вручну оператором судна:

- Пункт призначення.
- Очікуваний час прибуття.
- Осадка судна.

Інформація, яку приймає АІС

Система АІС приймає дані від інших суден і берегових станцій у зоні покриття. Отримувана інформація аналогічна тій, що передається:

- Ідентифікація судна (MMSI, назва, тип судна).
- Місцезнаходження та динамічні дані (координати, швидкість, курс).
- Пункт призначення та ЕТА.
- Інформація про небезпеку: сигнали лиха, об'єкти на шляху, зони обмежень

Статична і рейсова інформація передається з інтервалом – кожні 6 хвилин а також при зміні чи за запитом. Динамічна інформація передається в інтервалах приведених в таблиці 1.1 [3].

					171.6321M.KP.ПЗ.P1	Арк.
						13
Змн.	Арк.	№ документа	Підпис	Дата		

Таблиця 1.1 – Динамічна інформація

Стан судна	Інтервал передачі повідомлення
Судно на якорі або пришвартоване	3 хвилини
Судно рухається зі швидкістю ≤ 14 вузлів	10 секунд
Судно рухається зі швидкістю > 14 вузлів	6 секунд
Судно рухається зі швидкістю > 23 вузлів	2 секунди
Судно маневрує (курс змінюється швидко)	2 секунди

Наявність АІС на судах є обов'язковою вимогою Міжнародної конвенції з охорони людського життя на морі (SOLAS). Зокрема, глава V [4] цієї Конвенції встановлює мінімальні стандарти оснащення морських суден системами АІС. Крім того, деякі національні морські адміністрації можуть вимагати встановлення АІС на внутрішніх судах, які не підпадають під дію SOLAS.

Основні функції АІС на судах, що підпадають під дію SOLAS:

- Обмін інформацією між судном і береговою станцією: Сприяння ефективному управлінню рухом суден на перевантажених водних шляхах, шляхом обміну даними між судном і береговими службами, такими як служби управління рухом суден.
- Обмін інформацією між суднами: Забезпечення підвищення ситуаційної обізнаності шляхом автоматичного обміну даними про курс, швидкість, місцезнаходження та іншу важливу інформацію між суднами, що знаходяться в радіусі дії УКХ-зв'язку.

- Обмін інформацією про безпеку: Сприяння підвищенню безпеки на морі шляхом обміну інформацією про надзвичайні ситуації та інші події, що становлять загрозу безпеці судноплавства
- Автоматична звітність: Забезпечення автоматичної передачі інформації в зонах обов'язкової та добровільної звітності, що встановлюються національними морськими адміністраціями.

Міжнародна конвенція з охорони людського життя на морі (SOLAS) [4] є одним з найважливіших міжнародних договорів, що регулюють безпеку морського судноплавства. Прийнята в результаті низки трагічних морських катастроф, SOLAS встановлює мінімальні стандарти для проектування, будівництва, обладнання та експлуатації морських суден.

Судна, що підпадають під вимоги SOLAS, повинні бути оснащені транспондерами AIS класу А. Вони працюють у режимі трансляції SOTDMA та передають дані з потужністю 12,5 Вт. Динамічна інформація, така як місцезнаходження та курс, передається кожні дві-десять секунд під час руху та кожні три хвилини під час стоянки на якорі. Статична інформація та інформація, пов'язана з рейсом, наприклад, назва судна та вантаж, передається кожні шість хвилин. Транспондери АІС класу А повинні мати приймач DSC (156,525 МГц), зовнішній GPS, індикатор курсу та швидкості повороту, а також можуть передавати й отримувати текстові повідомлення, пов'язані з безпекою.

Судна, що не підпадають під вимоги SOLAS, можуть використовувати транспондери АІС класу В. Вони працюють у режимі трансляції CSTDMA та передають дані з потужністю два вати. Динамічні дані передаються кожні 30–180 секунд, а статичні дані - кожні шість хвилин. Приймач DSC та індикатор курсу не є обов'язковими. Передача текстових повідомлень, пов'язаних із безпекою, також не є обов'язковою і можлива лише за попереднього налаштування транспондера АІС класу В. Оскільки дані про

місцезнаходження оновлюються рідше, вони можуть бути менш точними для таких суден порівняно із суднами SOLAS.

1.2. УКХ канали АІС

АІС-трансивери використовують виділений для їх роботи радіочастотний діапазон, застосовуючи метод множинного доступу з часовим поділом (TDMA).

TDMA (Time Division Multiple Access) — це метод множинного доступу з розподілом за часом, який використовується для ефективного обміну даними між багатьма користувачами в одній частотній смузі. У цьому методі час ділиться на фіксовані інтервали, які називаються слотами, і кожному користувачу виділяється певний слот для передачі або прийому даних.

АІС зазвичай функціонує на двох паралельних каналах у морському УКХ-діапазоні 87В (161,975 МГц) і 88В (162,025 МГц), причому кожен з каналів поділяється між багатьма користувачами шляхом розбиття доступу на 2250 часових слотів за хвилину.

У стандартних TDMA-системах, таких як GSM, базова станція виступає в ролі контролера, що розподіляє часові слоти між користувачами. Проте АІС, яка працює у відкритому морі, не може покладатися на централізований елемент для управління слотами. Тому кожен АІС-трансивер самостійно визначає своє місце у часовому розподілі та уникає використання слотів, зайнятих іншими суднами в зоні прийому, щоб уникнути конфліктів передач.

Синхронізація часових слотів TDMA забезпечується за допомогою GPS, який надає точне загальне часове посилення для всіх АІС-трансиверів.

Перша схема TDMA, створена для АІС, отримала назву Самоорганізований Множинний Доступ з Часовим Розподілом (SOTDMA). Вона використовується в АІС-трансиверах класу А у всьому світі. Крім того,

					171.6321М.КР.ПЗ.Р1	Арк.
Змн.	Арк.	№ документа	Підпис	Дата		16

існують інші схеми доступу, які сумісні з SOTDMA і призначені для специфічних типів передач даних або для окремих категорій АІС-пристроїв.

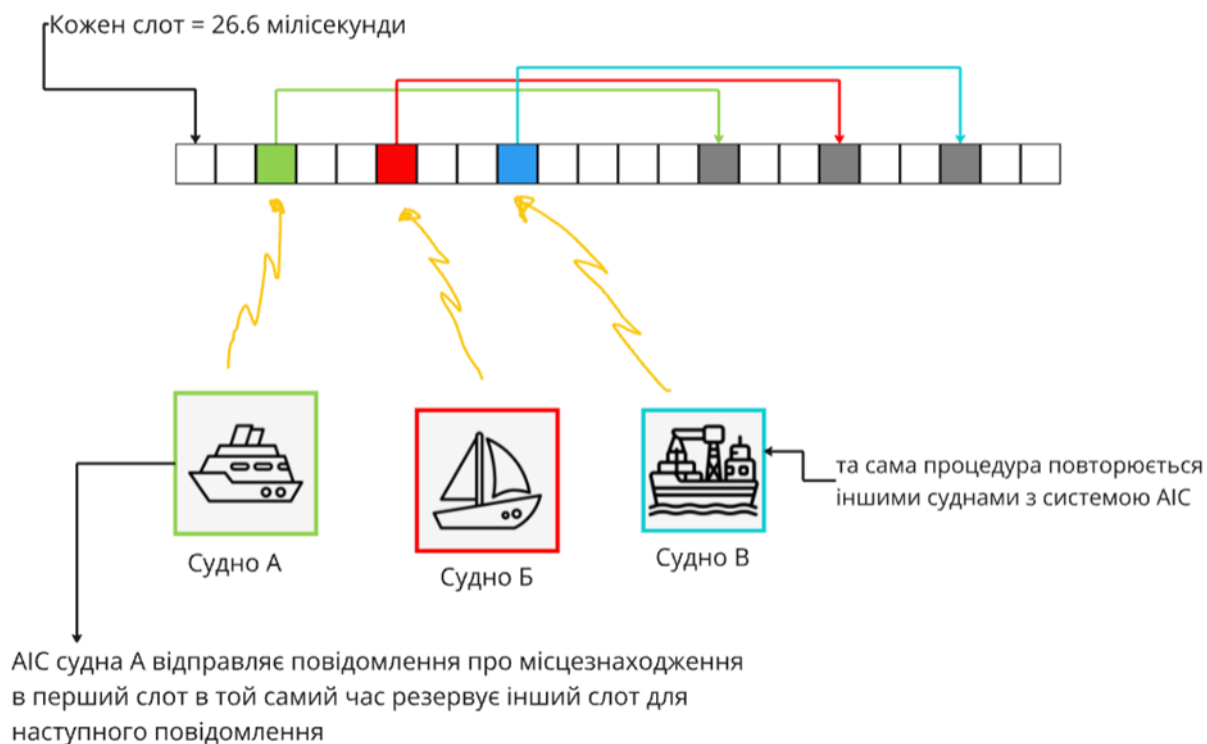


Рисунок 1.2 – Схема множинного часовим розподілом каналів

Схеми доступу TDMA, що використовуються в AIS [3]:

- SOTDMA – Самоорганізований Множинний Доступ із Часовим Розподілом;
- ITDMA – Інкрементний Множинний Доступ із Часовим Розподілом;
- CSTDMA – Множинний Доступ із Часовим Розподілом із Виявленням Несучої;
- RATDMA – Випадковий Множинний Доступ із Часовим Розподілом;
- FATDMA – Фіксований Множинний Доступ із Часовим Розподілом;
- Modified SOTDMA – Модифікований Самоорганізований Множинний Доступ із Часовим Розподілом (також відомий як RATDMA, Попередньо Оголошений Множинний Доступ із Часовим Розподілом).

SOTDMA є найскладнішою схемою доступу TDMA, розробленою для AIS, і слугує основою для автономної роботи мережі у відкритому морі

Основні принципи роботи SOTDMA [3] (рис. 1.2):

- Усі станції синхронізуються за спільним часовим еталоном (отриманим із GPS), що дає змогу точно визначати час початку кожного TDMA-слота.
- Кожна передача містить інформацію про слот TDMA, який станція використовуватиме для наступних передач. Це допомагає іншим станціям формувати «мапу» зайнятих слотів і визначати, які станції їх використовують.
- Станції уникають використання зайнятих іншими станціями слотів, для власних передач. Це запобігає конфліктам, коли дві станції в зоні досяжності намагаються одночасно використовувати один і той самий слот.
- Мобільні станції, переміщуючись у нові зони, зустрічають інші станції з відмінним розподілом слотів. У таких випадках вони змінюють свої виділення слотів, створюючи динамічну та самоорганізовану систему у просторі й часі.

SOTDMA також ефективно керує пропускнуою здатністю в перевантажених районах. Якщо всі слоти TDMA зайняті, вводяться правила «повторного використання слотів», які дозволяють задіяти слоти, зайняті найбільш віддаленими станціями. Це зменшує розмір «комірки» AIS, гарантуючи, що звіти про позиції найближчих суден, які критично важливі для безпеки навігації, залишаються доступними.

В зоні дії базової (берегової) станції призначення слотів для передачі виконує сама базова станція. Такий алгоритм називається FATDMA

FATDMA [3] – це схема доступу TDMA з ручним керуванням, у якій пристрої AIS попередньо налаштовані на використання певних TDMA-слотів

					171.6321M.KP.ПЗ.Р1	Арк.
						18
Змн.	Арк.	№ документу	Підпис	Дата		

для всіх передач. FATDMA використовується лише базовими станціями AIC та станціями AtoN AIC.

Станція AIC працює не тільки на двох каналах TDMA, які використовуються для передачі даних AIC, але також одночасно функціонує на каналі DSC (канал 70). Який використовується береговими станціями для динамічного призначення робочих каналів для AIC. Це дозволяє оптимізувати роботу мережі, враховуючи локальні умови радіообміну.

1.3. Особливості розповсюдження радіохвиль УКХ-діапазону

Ультракороткі хвилі з частотою понад 30 МГц не здатні відбиватися від іоносфери, тому їх поширення через іоносферні хвилі неможливе. Оскільки ці хвилі сильно поглинаються поверхнею Землі, їх розповсюдження земними хвилями в УКХ-діапазоні обмежене на короткі відстані [5].

У цьому діапазоні хвилі можуть поширюватися лише у формі прямих хвиль, тобто хвиль, що розповсюджуються в межах прямої видимості.

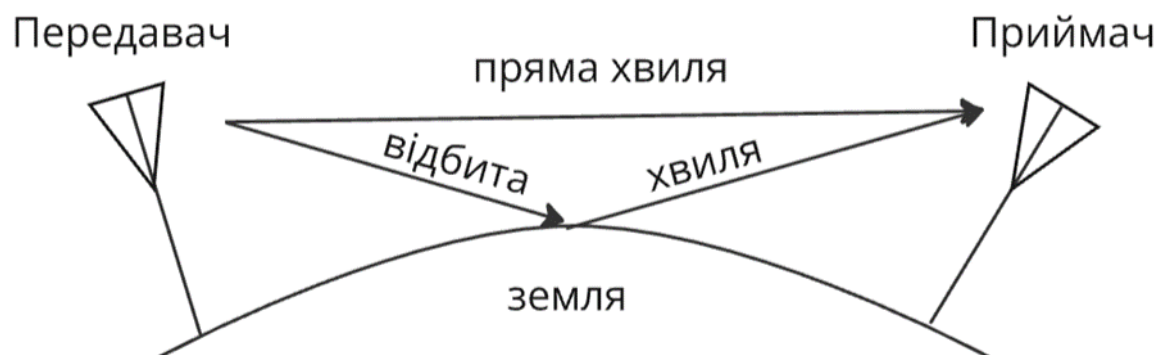


Рисунок 1.3 – Прямі та відбиті від землі хвилі при поширенні земної хвилі.

Пряма видимість (LOS) для радіозв'язку по наземному шляху завжди включає як пряму хвилю, так і хвилю, відбиту від земної поверхні(рис.1.3). Радіохвилі між передавачем (TX) і приймачем (RX) поширюються одночасно двома різними шляхами. Хоча в міських умовах часто зустрічаються

додаткові відбиті траєкторії, спричинені будівлями та іншими об'єктами, особливо для сигналів на ультракоротких хвилях та вищих частотах.

Припустімо, що сигнали досягають приймальної антени RX як по прямій траєкторії, так і по траєкторії, відбитій від земної поверхні, після чого ці хвилі комбінуються. Якщо обидва сигнали приходять в фазі, їх векторна сума збільшує амплітуду прийнятого сигналу. У випадку фазового зсуву відбувається деструктивна інтерференція, що призводить до зниження загальної сили сигналу. Таким чином, хоча рівень сигналу УКХ/УВЧ поступово зменшується зі збільшенням відстані між TX і RX, це зменшення не є плавним.

Зі збільшенням відстані до приймача RX сила сигналу змінюється нерівномірно, утворюючи своєрідний "хвилястий" профіль. Ця "нерівність" виникає через те, що різниця фаз між прямою та відбитою хвилями постійно змінюється з відстанню, що, у свою чергу, впливає на результуючу силу сигналу після їх векторного додавання.

Крім того, через кривизну Землі ефективність поширення сигналу в умовах LOS обмежується горизонтом. Відстань до горизонту визначається висотою антен TX та RX над поверхнею Землі. Чим вище розташовані антени, тим більша відстань до горизонту. Водночас, радіогоризонт зазвичай знаходиться значно далі, ніж оптичний горизонт.

Вплив тропосферних факторів на поширення радіохвиль VHF

Атмосферні умови, такі як хмарність, дощ, блискавка чи грім, мають мінімальний або незначний безпосередній вплив на поширення радіосигналів у діапазоні дуже високих частот (VHF). Зокрема, радіосигнали в діапазоні VHF та нижньому діапазоні ультрависоких частот (UHF) демонструють високу стійкість до таких факторів, як поглинання дощем або хмарами. Хмари практично не впливають на наземні радіокомунікаційні системи, тоді як блискавки можуть лише спричинити переривчасті імпульсні перешкоди. У

					171.6321M.KP.ПЗ.Р1	Арк.
Змн.	Арк.	№ документа	Підпис	Дата		20

загальному випадку передавання сигналів у діапазоні VHF залишається незмінним навіть за наявності цих погодних явищ.

Проте це не означає, що погодні умови, наприклад, дощ чи хмарність, абсолютно не впливають на поширення радіосигналів VHF. Їхній вплив радше опосередкований. Наприклад, зміна вертикального профілю вологості в тропосфері може впливати на поведінку сигналів. У спекотний день після дощу випаровування із нагрітої земної поверхні може спричинити підвищення парціального тиску водяної пари на висотах до 100 – 200 метрів. Це здатне змінити характеристики поширення наземних радіосигналів VHF-UHF.

Тропосферні умови значною мірою визначають поширення VHF/UHF сигналів у зоні прямої видимості. Зокрема, під час проходження крізь тропосферу радіохвилі можуть змінювати свою траєкторію, що супроводжується вигином їхнього шляху. Напрямок і ступінь цього вигину відіграють ключову роль у розширенні або обмеженні зони покриття зв'язку.

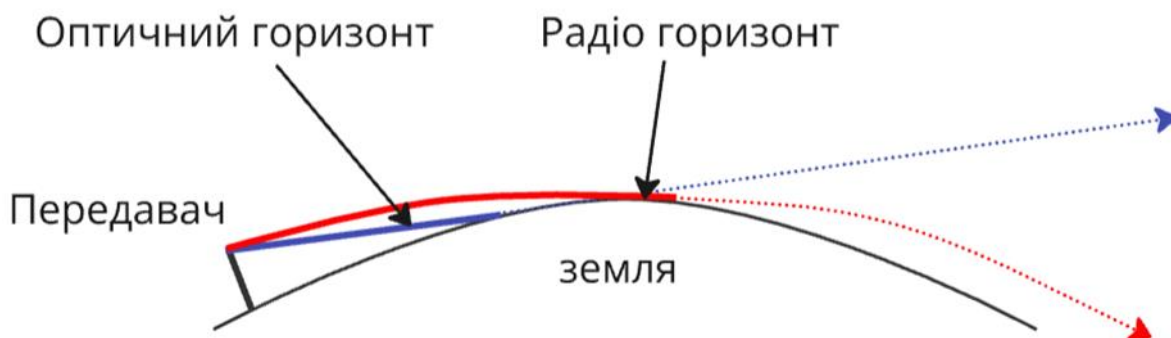


Рисунок 1.4 – Візуальне представлення типової дальності радіозв'язку за межами оптичного горизонту для радіочастот VHF та UHF в стандартних атмосферних умовах.

Без вигину сигнал прямої видимості через сферичну форму Землі досягає лише оптичного горизонту, після чого прямує у відкритий космос. У таких умовах неможливо забезпечити зв'язок за межами лінії прямої

видимості. Однак завдяки фізичним явищам, що відбуваються в атмосфері, траєкторія VHF/UHF сигналів часто вигинається вниз, уздовж кривизни Землі. Це явище дозволяє суттєво розширити дальність зв'язку (рис.1.4).

Втім, існують також небажані сценарії, коли сигнали вигинаються вгору, що значно скорочує дальність зв'язку і може навіть унеможливити його досягнення оптичного горизонту. В таких випадках визначальними є атмосферні умови, що створюють відповідний градієнт рефракції. Таким чином, поширення радіохвиль VHF/UHF суттєво залежить від вигину їхніх траєкторій, що обумовлено особливостями тропосфери.

Явище вигину VHF/UHF сигналів можна порівняти з рефракцією HF-сигналів в іоносфері. У випадку HF радіохвилі відхиляються від своєї початкової траєкторії через градієнт показника заломлення, спричинений варіацією плазмової щільності в іоносферних шарах. Подібним чином у тропосфері радіохвилі VHF вигинаються через вертикальний градієнт показника заломлення, що виникає внаслідок змін щільності повітря на різних висотах.

Вертикальний градієнт щільності повітря є наслідком атмосферних параметрів (рис.1.5), таких як:

- Температура повітря — зміна температури з висотою впливає на щільність повітря та створює градієнт.
- Атмосферний тиск — його зменшення з висотою призводить до зменшення щільності повітря.
- Вологість повітря — зміни концентрації водяної пари також впливають на загальну щільність.

Усі ці параметри у комплексі формують умови, за яких виникає градієнт рефракції. Від його величини залежить ступінь вигину траєкторії сигналу[6].

					171.6321М.КР.ПЗ.Р1	Арк.
						22
Змн.	Арк.	№ документу	Підпис	Дата		

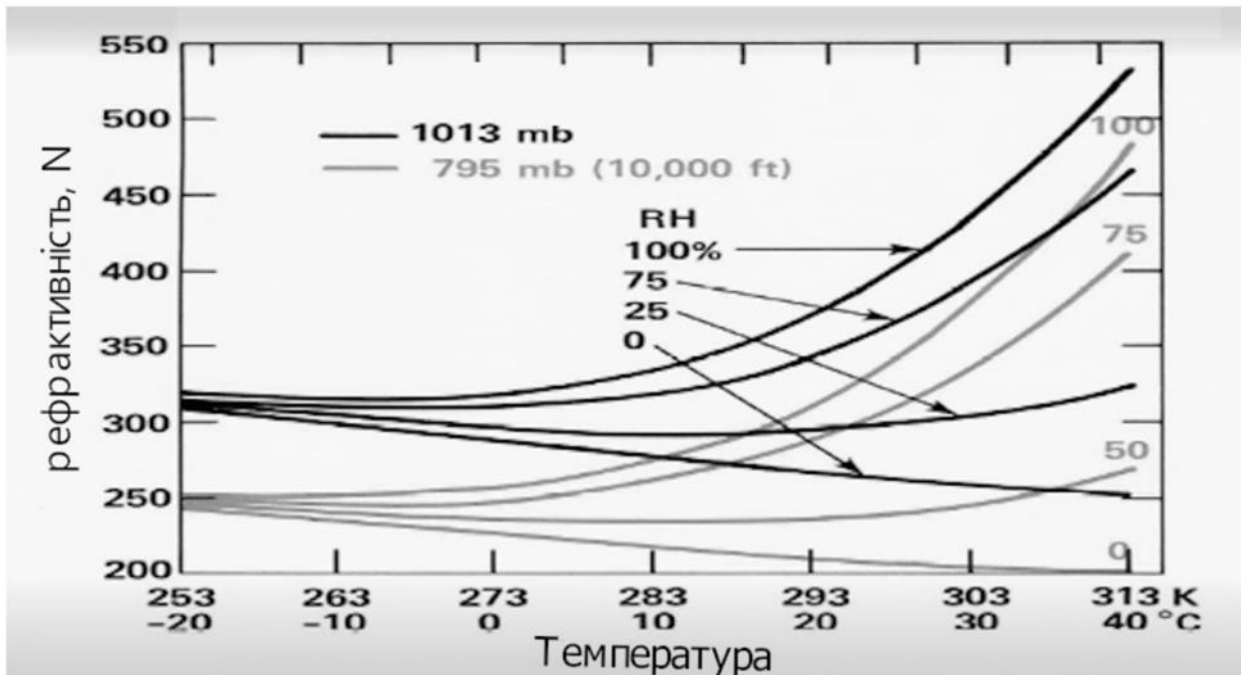


Рисунок 1.5 – Типова зміна рефракційної здатності (N) стовпа повітря в умовах стандартної атмосфери на основі зміни тиску, температури та вологості

Щільність повітря важлива, оскільки швидкість поширення електромагнітних хвиль (зокрема світла, теплового випромінювання або радіохвиль) є обернено пропорційною до щільності середовища. У щільнішому повітрі швидкість радіохвиль зменшується, а у розрідженому - збільшується. Максимальна швидкість радіохвиль спостерігається у вакуумі. Хоча різниця швидкості поширення радіосигналу у вакуумі та в атмосфері незначна, вона достатня для виникнення іншого фізичного явища - рефракції.

Рефракція відбувається, коли змінюється щільність середовища, а отже, і швидкість поширення хвиль. Величина рефракції характеризується показником заломлення (n) або рефрактивністю (N). На межі двох середовищ із різною щільністю (наприклад, повітря і води) виникає різниця у показниках рефракції, що спричиняє відхилення радіохвиль від прямолінійного шляху на границі середовищ.

Однак не всі межі заломлення є різкими, як у випадку неоднорідних середовищ, описаному вище. Середовище може бути однорідним, наприклад,

вертикальний стовп повітря, але який є змінним за градієнтом щільності вздовж вертикалі. Це, у свою чергу, призводить до поступової зміни градієнта показника заломлення. Отже, радіохвиля, що поширюється під кутом до такого стовпа повітря, зазнаватиме постійного заломлення та поступового викривлення своєї траєкторії за криволінійною формою.

У нормальних умовах щільність повітряного стовпа над поверхнею Землі поступово зменшується з висотою через зниження температури, вологості та атмосферного тиску. Це, у свою чергу, спричиняє зменшення показника заломлення з висотою, формуючи градієнт рефракції.

Градієнт рефракції визначається як зміна рефрактивності ΔN відносно зміни висоти ΔH . Градієнт рефракції $= \Delta N / \Delta H$. Ця величина зазвичай вимірюється у ΔN на кілометр висоти ($\Delta N/\text{км}$).

За нормальних метеорологічних умов рефрактивність повітряного стовпа зменшується з висотою. Таким чином, величина ΔN є від'ємною, а градієнт рефракції також матиме від'ємне значення. У середньому градієнт рефракції в тропосфері за стандартних умов становить приблизно $-39N/\text{км}$. Це значення прийнято як стандартне і використовується для характеристики нормального стану атмосфери, який називається стандартною атмосферою.

1.4. Типові градієнти атмосферної рефракції та поширення радіохвиль у діапазоні УКХ

Стандартна рефракція (рис.1.6) - це стан атмосфери, який переважає у більшості регіонів світу за помірних і середньостатистичних погодних умов. Цей стан характеризується стандартним градієнтом рефракції, що становить $-39N/\text{км}$, як було визначено раніше. Однак градієнти рефракції, що наближаються до цього значення в діапазоні від $-79N/\text{км}$ до $0N/\text{км}$, також класифікуються як умови майже стандартної рефракції.

За таких умов радіосигнали у діапазоні VHF-UHF зазнають поступового, але незначного вигину у напрямку до поверхні Землі. Хоча

					171.6321M.KP.ПЗ.Р1	Арк.
						24
Змн.	Арк.	№ документу	Підпис	Дата		

величина цього вигину є малою, вона забезпечує додаткове розширення зони покриття радіосигналу за межі оптичного горизонту. Ступінь цього розширення визначається градієнтом рефракції, що дозволяє ввести поняття ефективного коефіцієнта радіуса Землі, позначеного як K [7].

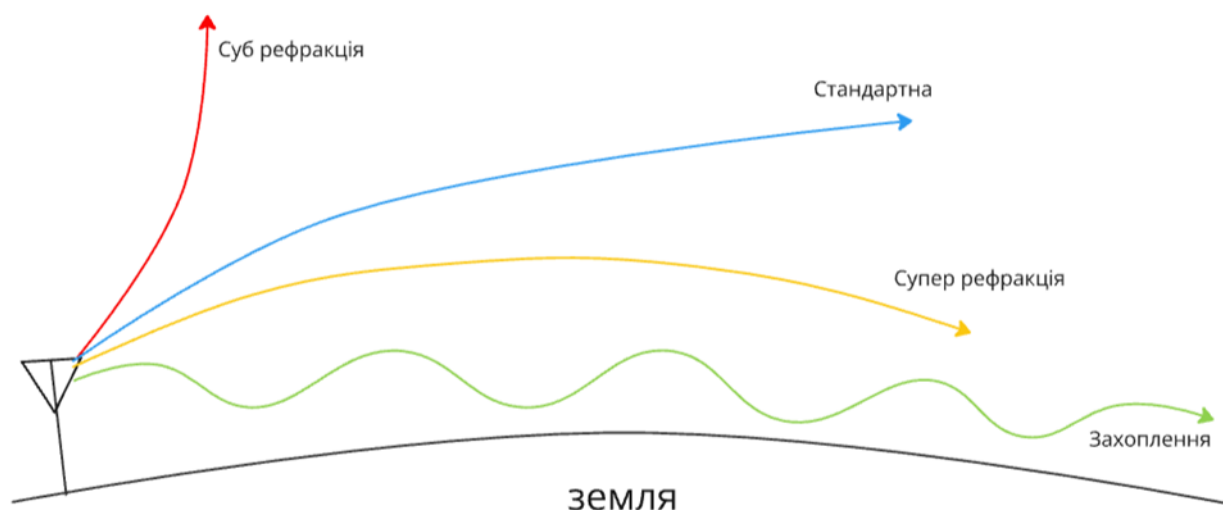


Рисунок 1.5 – Типове прояв умов атмосферної (тропосферної) рефракції

В умовах стандартної атмосфери зона покриття радіосигналу розширюється приблизно на $\sqrt{4/3}$ відстані оптичного горизонту. Це відповідає ефективному коефіцієнту радіуса Землі $K = 4/3$. Таким чином, радіогоризонт у середньому знаходиться на $\sqrt{4/3}$, або приблизно на 15,5% далі, ніж оптичний горизонт.

Суперрефракція [7] (рис. 1.6) - це явище, за якого градієнт рефракції вертикального стовпа повітря зазвичай знаходиться в межах від -157 N/км до -79 N/км. Як зазначалося раніше, стандартна атмосфера характеризується градієнтом рефракції до значення -79 N/км. Це значення є граничним, після якого починаються умови суперрефракції. У міру збільшення від'ємності цього градієнта поширювана радіохвиля зазнає все більшого вигину вниз у напрямку до поверхні Землі.

Через збільшення кривини траєкторії поширення сигналу зона покриття поширення радіосигналів перевищує ту, що спостерігається за умов стандартної атмосфери, де кривина траєкторії менша. Іншими словами, явище суперрефракції спричиняє значне розширення наземного покриття сигналів VHF у порівнянні з очікуваним за нормальних умов.

Із посиленням негативного градієнта рефракції кривина траєкторії сигналу поступово збільшується. Наприклад, радіус покриття радіосигналу в діапазоні VHF, який за умов стандартної рефракції становить приблизно 50 км, може легко розширитися до кількох сотень кілометрів за умов сильної суперрефракції. Такі умови можуть бути тимчасовими, тривати кілька годин або, у рідкісних випадках, навіть декілька днів поспіль. Розуміння цього явища є критично важливим для розрахунків у радіозв'язку та прогнозування поведінки сигналів у різних атмосферних умовах.

Магічний градієнт рефрактивності [7] (-157N/км) - одним із крайніх значень діапазону суперрефракції є градієнт рефрактивності -157N/км , який створює унікальний фізичний ефект. Особливість цього значення полягає в тому, що кривина траєкторії горизонтально поширюваної радіохвилі точно збігається з кривиною поверхні Землі. Іншими словами, радіосигнал поширюється паралельно до поверхні Землі, зберігаючи постійну відстань від неї по всій довжині свого шляху навколо сферичної Землі.

Якщо градієнт рефрактивності стане навіть трохи менш негативним за -157N/км , кривина траєкторії сигналу буде меншою за кривину Землі, і сигнал поступово відхилиться вгору, гублячись у просторі. Навпаки, якщо градієнт стане більш негативним за -157N/км , радіосигнал вигнеться сильніше і досягне поверхні Землі через певну відстань.

При градієнті -157N/км сигнал поширюється точно паралельно до поверхні Землі і не зустрічає горизонту. За таких умов горизонт для радіосигналу просто не існує. Така траєкторія сигналу нагадує модель поширення на гіпотетично пласкій Землі.

					171.6321М.КР.ПЗ.Р1	Арк.
						26
Змн.	Арк.	№ документа	Підпис	Дата		

Якби магічний градієнт рефрактивності міг утримуватися стабільно на великих відстанях навколо Землі, це зробило б можливим передачу радіосигналів у діапазоні VHF на надзвичайно великі відстані. Однак у реальності це малоймовірно. Динамічна природа атмосфери та постійно змінювані погодні умови не дозволяють значенню $-157\text{N}/\text{км}$ залишатися стабільним і постійним протягом тривалого часу.

Навіть за умови тимчасового досягнення цього значення градієнта воно рідко спостерігається на достатньо великих територіях. Таким чином, магічний градієнт рефрактивності є скоріше предметом теоретичного інтересу, аніж практичним явищем для забезпечення тривалих радіозв'язків

Умови захоплення сигналу [7] (рис. 1.6) - якщо градієнт рефрактивності стає більш негативним за $-157\text{N}/\text{км}$, радіосигнал у діапазоні VHF починає ще сильніше згинатися вниз. Його траєкторія набуває кривини, що перевищує кривину Землі. У результаті траєкторія сигналу набуває форми інверсної параболи: сигнал продовжує згинатися вниз, поки не досягне поверхні Землі. Після цього відбувається його відбиття від поверхні, і він продовжує поширення. Такий процес може повторюватися багаторазово, формуючи каскад відбиттів, що дозволяє сигналу долати надзвичайно великі відстані.

Цей механізм нагадує іоносферне поширення сигналів у діапазоні HF, де сигнал згинається під дією іоносфери. Основна відмінність полягає в тому, що при іоносферному поширенні вигин сигналу відбувається на висотах сотень кілометрів над Землею. У випадку захоплення в тропосфері сигнал у діапазоні VHF-UHF згинається на значно менших висотах — зазвичай до 10 км, а найчастіше між поверхнею Землі та висотою близько 3 км. Основною причиною такого явища є високо негативний градієнт рефрактивності вертикального стовпа повітря над поверхнею Землі.

Явище захоплення найчастіше спостерігається в прибережних районах, але може також виникати й у внутрішніх регіонах. У континентальних

					171.6321М.КР.ПЗ.Р1	Арк.
						27
Змн.	Арк.	№ документа	Підпис	Дата		

умовах захоплення, як правило, відбувається після значних опадів, особливо літнього або осіннього дощу, коли атмосферні умови над поверхнею стають помітно вологими.

Субрефракція (рис.1.6) [7] - атмосферні умови, що характеризуються позитивним градієнтом рефракції, як правило, негативно впливають на якість радіозв'язку у VHF-діапазоні. Вони призводять до зменшення дальності зв'язку, яка може стати навіть меншою за оптичний горизонт.

Граничним значенням між нормальними (стандартними) умовами поширення радіохвиль і субрефракцією є нульовий градієнт рефракції (0 N/км). За таких умов радіосигнал поширюється прямолінійно. Однак, коли градієнт рефракції набуває додатного значення, радіохвилі починають викривлятися вгору.

Субрефракція є значною проблемою для радіозв'язку у VHF-діапазоні, оскільки обмежує дальність передачі радіосигналу. Ці умови часто виникають за спекотної та сухої погоди, особливо в пустельних або кам'янистих регіонах з низьким рівнем вологості ґрунту.

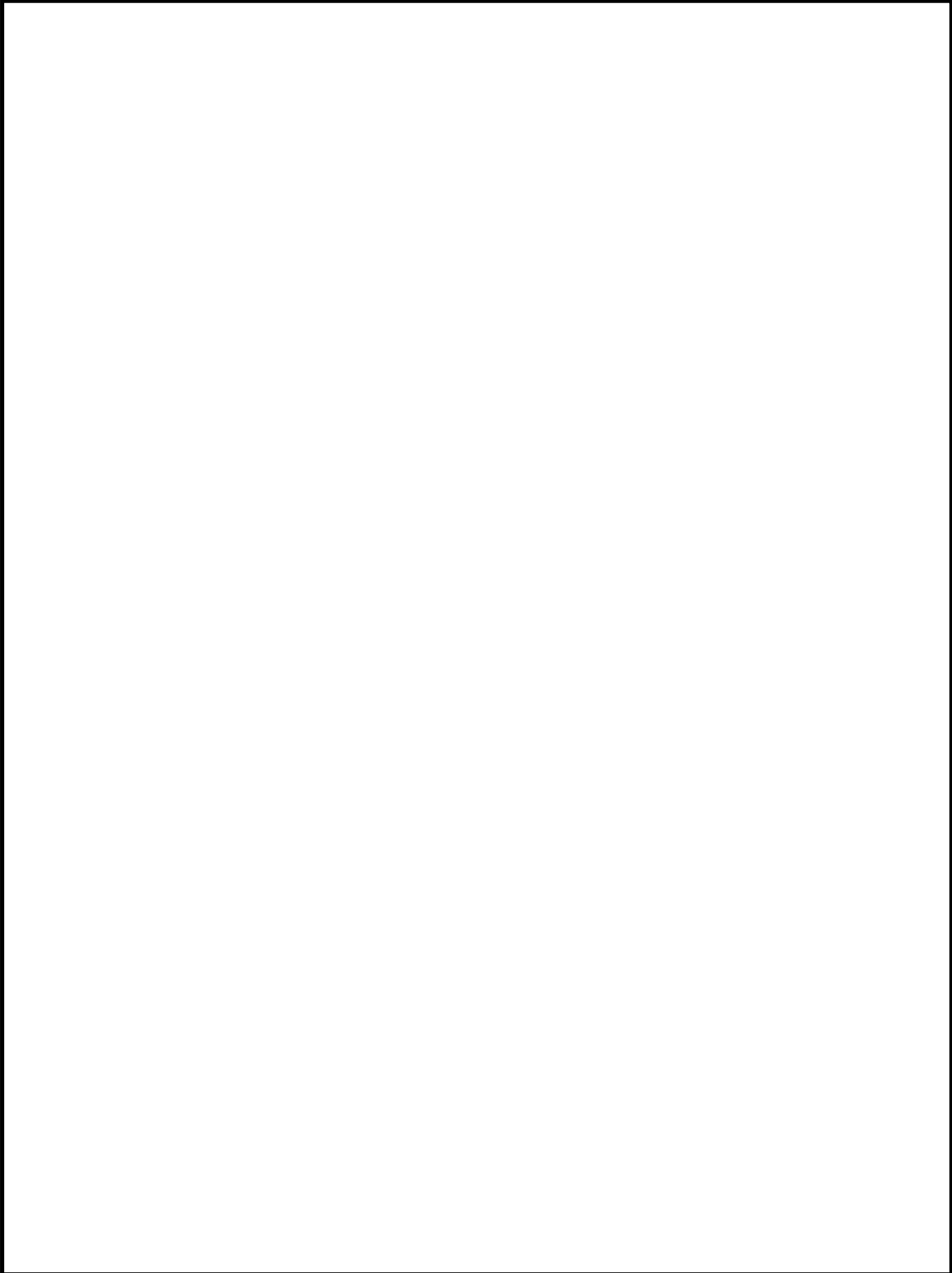
					171.6321М.КР.ПЗ.Р1	Арк.
						28
Змн.	Арк.	№ документу	Підпис	Дата		

Висновок

У даному розділі було проведено аналіз автоматизованої ідентифікаційної системи (AIS) як ключового елемента сучасних засобів морської навігації та безпеки. Розглянуто основні характеристики ультракороткохвильового (УКХ) діапазону, який використовується для передачі даних в AIS, та його переваги у забезпеченні стійкого зв'язку. Особливу увагу приділено схемам доступу TDMA, які є основою для ефективного розподілу частотного ресурсу серед користувачів

Крім того, було детально проаналізовано особливості розповсюдження УКХ-хвиль, включаючи вплив тропосферних факторів, таких як температурні градієнти, вологість та інші атмосферні явища. Встановлено, що ці фактори можуть суттєво впливати на якість зв'язку, спричиняючи явища рефракції, затухання та розсіювання сигналу.

					171.6321М.КР.ПЗ.Р1	Арк.
						29
Змн.	Арк.	№ документу	Підпис	Дата		



					171.6321М.КР.ПЗ.Р2			
Змн.	Арк.	№ документа	Підпис	Дата				
Розробник		Вишневський В. В.			Аномальне поширення УКХ-радіохвиль. Вплив на роботу АІС	Літ.	Арк.	Аркуші
Консультант							30	
Н. контр.		Добрінова Л. П.				НУК імені адмірала Макарова		
Керівник		Дьяконов О. С.						
Зав. каф.		Рябенський В. М.						

РОЗДІЛ 2. АНОМАЛЬНЕ ПОШИРЕННЯ УКХ РАДІОХВИЛЬ. ВПЛИВ НА РОБОТУ АІС

2.1. Аномальні профілі градієнтів рефрактивності для утворення хвилеводів радіосигналів у діапазоні УКХ

Впливи на поширення радіосигналів у діапазоні VHF, зумовлені різними умовами градієнтів рефрактивності атмосфери, такими як стандартна рефракція, суперрефракція, субрефракція тощо. Усі ці явища обумовлені локальними погодними умовами в спокійному стані тропосфери.

Однак можливі ситуації, коли погодні явища в сусідніх регіонах взаємодіють із локальними погодними системами та іноді домінують над ними, спричиняючи виникнення аномальних умов у вертикальному стовпі атмосфери. Коли такі атмосферні фронти підходять до локальної системи та взаємодіють із нею, це може призводити до створення значно неоднорідних вертикальних градієнтів рефрактивності. Подібні взаємодії між двома атмосферними фронтами викликають стратифікацію повітряного стовпа тропосфери. Наприклад, теплий повітряний шар із одного регіону на певній висоті може проникати в холодніший шар атмосфери тієї ж висоти в сусідньому регіоні. Це призводить до утворення теплого шару повітря, який "запечатується" між холодними шарами зверху та знизу. Таке явище отримало назву стратифікованого градієнта рефрактивності [6].

Варто зазначити, що не завжди в атмосферу надходить теплий шар повітря. Навпаки, іноді прохолодне повітря може проникати у вертикальний градієнт. Висоти над рівнем землі, на яких виникають подібні стратифікації, можуть значно варіюватися – від поверхні землі до кількох кілометрів. Товщина цих аномальних шарів також змінюється в широких межах, від кількох десятків до сотень метрів.

Наприклад, теплий шар повітря, що надходить на певній висоті з сусідньої області, може порушити однорідний вертикальний профіль рефрактивності, який існував раніше. Унаслідок цього утворюється явище,

					171.6321M.KP.ПЗ.P2	Арк.
						31
Змн.	Арк.	№ документа	Підпис	Дата		

яке називають термічною інверсією. На певній висоті формується чіткий шар із більш значним негативним градієнтом рефрактивності. Це явище може спричинити захоплення радіосигналів у діапазонах VHF та UHF у межах цього інверсійного шару. Таким чином формується невидимий канал – або "хвилевод" – на певній висоті над землею.

Радіосигнали VHF, які потрапляють у цей хвилевод, залишаються "захопленими" у ньому та можуть поширюватися на дуже великі відстані. Поширення сигналу в межах таких хвилеводів супроводжується мінімальними втратами енергії. До тих пір, поки умови термічної інверсії залишаються стабільними, явище хвилеводного поширення радіосигналів VHF-UHF зберігається.

Явище атмосферного хвилеводу, в радіозв'язку, може виникати різними способами. Залежно від розташування хвилеводів, їх класифікують як піднятий (elevated) хвилевід або поверхневий (surface) хвилевід.

Такі хвилеводи не мають однакових характеристик. Деякі з них можуть бути оптимально придатними для діапазону VHF, тоді як інші краще підходять для UHF, але неспроможні підтримувати поширення сигналів VHF. Характеристики поведінки хвилеводів визначаються різними комбінаціями атмосферних параметрів. Як загальне правило, середня товщина шару градієнта термічної інверсії (висота, на якій утворюється хвилевід) значною мірою визначає його здатність забезпечувати поширення сигналів.

Наприклад, товщина хвилеводу приблизно 200 метрів може бути достатньою для підтримки поширення радіосигналів діапазону 2 метри (VHF), тоді як для захоплення та поширення сигналів діапазону 70 см (UHF) цілком достатньо шару товщиною близько 90 метрів. Проте такий тонкий хвилевід не зможе підтримувати поширення сигналів діапазону 2 метри (VHF).

Ці особливості застосовуються як до піднятих хвилеводів, так і до поверхневих. Вибір діапазону частот, що можуть ефективно поширюватися в

					171.6321M.KP.ПЗ.P2	Арк.
						32
Змн.	Арк.	№ документа	Підпис	Дата		

конкретному хвилеводі, значною мірою залежить від його фізичних характеристик та атмосферних умов.

Піднятий хвилевод [7, 8] - на рис.2.1 зображено типовий сценарій припіднятого повітряного каналу.

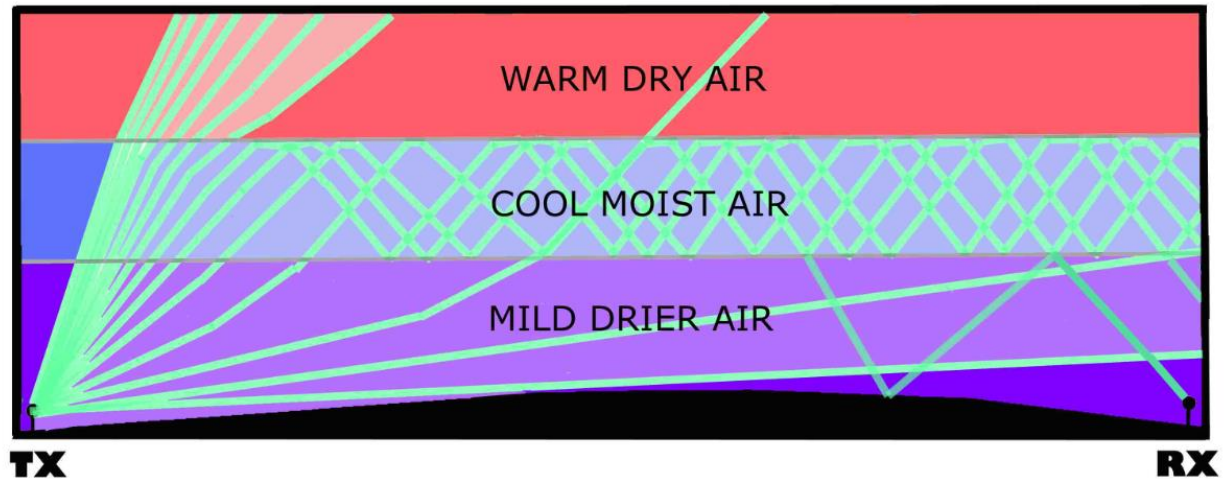


Рисунок 2.1 – Піднятий тропосферний хвилевод

Це явище викликається шаром більш теплого повітря, що рухається з сусіднього регіону та вступає в контакт з існуючим повітряним стовпом. Такий шар теплого повітря може створювати стратифікацію на значній висоті від земної поверхні. Зазвичай більшість припіднятих інверсій, які утворюють повітряні канали, спостерігаються на висотах близько 3-5 км над землею, хоча в окремих випадках такі канали можуть формуватися і на висотах від 1 до 10 км.

Припідняті хвилеводи ефективно поширюють радіосигнали в діапазоні VHF, як показано на ілюстрації. Радіохвиля, що поширюється, захоплюється або направляється в зону, що знаходиться вище земної поверхні. Оскільки сигнал відбивається між межами повітряного каналу, він може поширюватися на дуже великі відстані. Втрата сигналу в межах припіднятого повітряного каналу є мінімальною і підпорядковується, зокрема, закону зворотних квадратів для поширення вільним простором.

Такі канали можуть утворюватися в різних географічних регіонах за різних умов, але вони часто спостерігаються в долинах, де теплий повітряний шар з навколишніх високогірних регіонів може проникати в долину. Це повітря зазвичай тепліше, ніж повітря в самій долині. Повітря, що рухається поблизу поверхні землі з навколишніх високих територій, нагрівається внаслідок більш теплої земної поверхні. Коли цей повітряний шар потрапляє в долину, він формує теплий стратифікований шар у вертикальному повітряному стовпі над долиною, створюючи умови для утворення припіднятого хвелеводу.

Якщо детальніше розглянути область інверсії, показану для припіднятого повітряного каналу, можна помітити, що нижня частина теплового повітряного шару утворює верхню межу цього каналу. Нижня межа каналу знаходиться набагато нижче в межах вертикального повітряного стовпа, який природним чином тепліший через свою висоту над землею. Повітряний канал сам по собі є холоднішою зоною, що розташована між природно теплими шарами зверху та знизу.

Поверхневі хвелеводи (рис.2.2) [7, 8] демонструють схожі характеристики з класичними умовами захоплення сигналу, які викликані градієнтом рефракції поблизу поверхні землі. Це явище спричинене природним градієнтом рефракції, що є більш негативним, ніж -157 N/км . Утворення таких умов може вимагати наявності специфічних локальних погодних факторів, зокрема високої вологості біля поверхні, що сприяє підвищенню парціального тиску водяної пари. Утворення поверхневих каналів може бути обумовлене не лише температурним градієнтом, але й достатнім градієнтом вологості, який створює необхідні умови. Як зазначалося раніше, температура, вологість (тиск водяної пари) і атмосферний тиск є основними факторами, що впливають на формування градієнтів рефракції, які призводять до виникнення хвелеводів та інших подібних явищ.

					171.6321M.KP.ПЗ.P2	Арк.
						34
Змн.	Арк.	№ документа	Підпис	Дата		

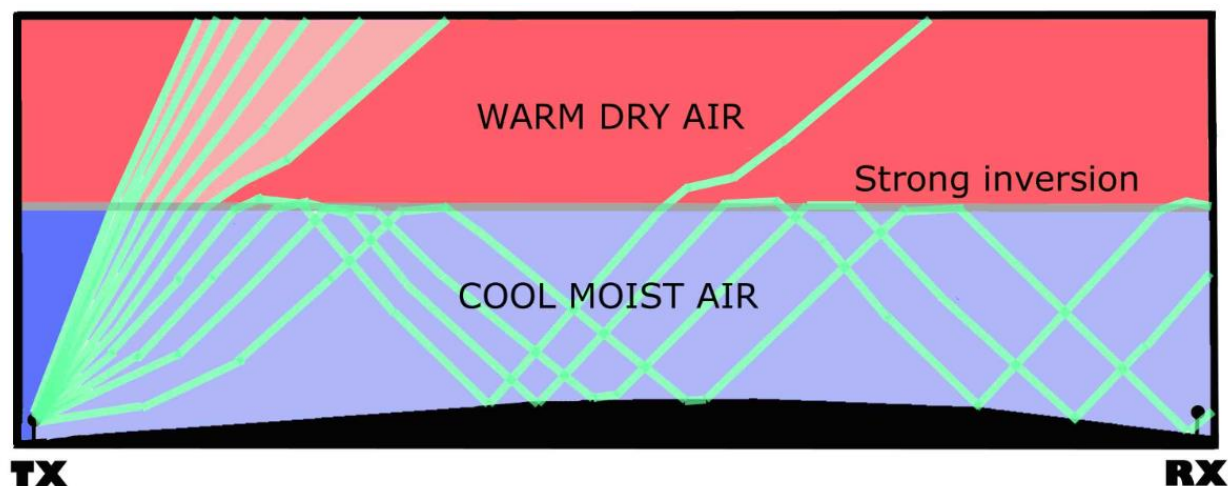


Рисунок 2.2 – Поверхневий хвиевод

Головна відмінність між утворенням поверхневих каналів і локальними умовами захоплення полягає у масштабі цих явищ. Якщо локальні канали формуються під впливом обмежених місцевих погодних умов, то поверхневі канали виникають внаслідок взаємодії локальних атмосферних умов із погодними фронтами сусідніх регіонів.

Утворення поверхневих хвиеводів подібне до формування піднятих хвиеводів, але з деякими важливими відмінностями. Найбільша різниця полягає у висоті шару інверсії. Для поверхневих каналів ця висота зазвичай становить 300–500 м над рівнем землі, але не перевищує 1 км. Через таку низьку висоту інверсійного шару, простір для формування нижньої межі каналу відсутній. Таким чином, поверхня землі виступає як нижня межа каналу, а верхню межу утворює тепловий інверсійний шар. Це пояснює, чому такі канали називають поверхневими.

Поверхневі канали характеризуються більшими втратами сигналу порівняно з піднятими каналами через значне затухання сигналу під час кожного відбиття від земної поверхні. До цього додаються інші фізичні явища, такі як поглинання та розсіювання хвиль поверхнею землі, що збільшують загальне затухання сигналу.

					171.6321M.KP.ПЗ.P2	Арк.
						35
Змн.	Арк.	№ документа	Підпис	Дата		

Проте поверхневі канали мають певні переваги. Вони частіше використовуються в аматорській радіозв'язку, оскільки їхнє формування є більш передбачуваним і регулярним у порівнянні з піднятими каналами. У той час як підняті канали частіше зустрічаються на високих частотах УВЧ і мікрохвиль, вони рідше використовуються на частотах УКХ.

Хоча найчастіше утворення каналів пояснюється появою теплого шару повітря, що зміщується із сусіднього регіону, це не єдиний можливий механізм. Наприклад, холодний шар повітря, що формується над морем чи великим озером, може переміщуватися до прибережної зони протягом дня, коли температура повітря над суходолом вища. Цей щільний, вологий шар повітря витісняє тепліший, менш щільний повітряний шар над суходолом, створюючи умови низьковисотної інверсії, які сприяють формуванню надійного поверхневого хвильоводу.

Випаровувальний хвильовод [7, 8] є своєрідним різновидом поверхневого хвильоводу, однак його варто розглядати окремо через специфічні особливості, зокрема явища, які сприяють його утворенню. На відміну від звичайних поверхневих хвильоводів, випаровувальні канали не формуються внаслідок взаємодії сусідніх атмосферних фронтів. Це локальне явище.

Хоча випаровувальні хвильоводи технічно можуть виникати і над сушею, наприклад, після дощів у спекотних тропічних регіонах, їх основна сфера поширення – це великі водойми, такі як моря, океани або великі озера.

Одним із ключових факторів, які впливають на рефракцію радіохвиль у повітрі, є вологість. Вона визначається парціальним тиском водяної пари, який зростає зі збільшенням вологості. Вищий тиск водяної пари спричиняє більшу рефракцію радіохвиль.

У теплі сонячні дні над поверхнею великих водойм інтенсивно випаровується вода, утворюючи насичений водяною парою шар атмосфери, що може сягати висоти понад 100 метрів. У цьому шарі спостерігається

					171.6321М.КР.ПЗ.Р2	Арк.
						36
Змн.	Арк.	№ документа	Підпис	Дата		

високий градієнт рефракції у порівнянні зі значно сухішим повітрям на вищих висотах. Такий градієнт створює сильну негативну рефракцію, яка зберігається на висотах до 100 метрів і вище.

Канал розповсюдження радіохвиль, що формується внаслідок зазначеного явища, називають випаровувальним хвилеводом. Це переважно морське явище, яке забезпечує наддалекий зв'язок поза межами прямої видимості у діапазоні УКХ. Радіопокриття на відстанях до 1000 км і більше стає можливим завдяки багаторазовому відбиттю радіохвиль між поверхнею води та шаром інверсії, що може знаходитися на висоті всього 100 м.

2.2. Деякі проблеми, пов'язані з тропосферними хвилеводами

Рисунок 2.3 допоможе пояснити процеси, що відбуваються в області прикордонного шару теплової інверсії, зображеного червоною лінією. На рисунку (2.3, а) видно, що радіосигнали, які досягають області інверсії під малими кутами, залишаються на тому боці кордону, де вони виникли. Сигнали, що формуються під шаром інверсії (наприклад, поблизу земної поверхні), поширюються вперед, одночасно заломлюючись донизу. Натомість сигнали, які генеруються вище за шар інверсії (наприклад, від літаків), заломлюються вгору, відштовхуючись від кордону, що знаходиться нижче них. При цьому сигнали, які досягають інверсії під малими кутами падіння, не можуть легко проникнути на іншу сторону прикордонного шару.

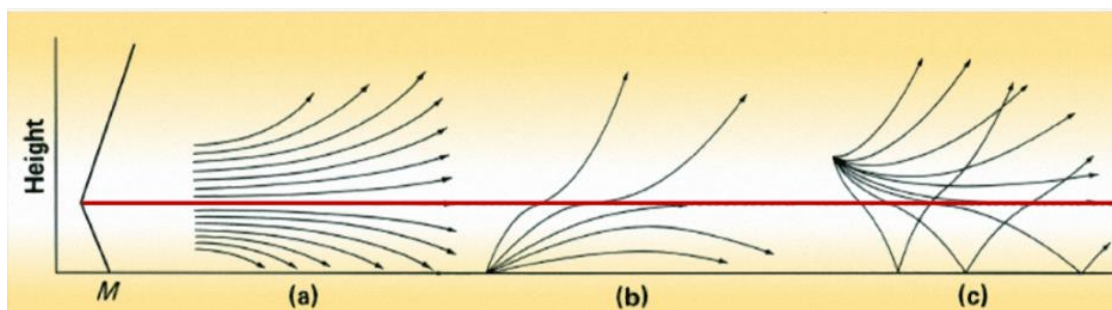


Рисунок 2.3 – Вплив шару теплової інверсії на характеристики вигину радіохвиль у діапазонах VHF-UHF.

Рисунок (2.3, b) та рисунок (2.3, c) додатково роз'яснюють цей принцип. У той час як сигнали під малими кутами не можуть перетнути межу теплової інверсії, сигнали під великими кутами мають змогу подолати цей бар'єр. Таке явище може спричиняти низку проблем, з якими стикаються системи зв'язку та радіолокації.

Основні проблеми, спричинені тропосферними хвилеводами:

- Проблеми зі зв'язком між супутниками та наземними станціями - умови сильного тропосферного хвилеводу можуть значно погіршувати якість сигналів між супутниками на низькій навколоземній орбіті (LEO) та наземними станціями, особливо коли супутники знаходяться поблизу горизонту. Через труднощі з проникненням сигналів через кордон інверсії на малих кутах, якість зв'язку різко знижується.
- Збої у зв'язку між наземними станціями та повітряними об'єктами - зв'язок між наземною станцією та літаком, що знаходиться на великій відстані або висоті, може ускладнюватися через ефекти повітряних каналів. Це призводить до зменшення дальності зв'язку та погіршення співвідношення сигнал/шум (SNR).
- Вплив на роботу радіолокаційних систем - тропосферні повітряні канали можуть знижувати ефективність радарів. Дальність виявлення об'єктів скорочується через значне ослаблення сигналів, які намагаються проникнути через кордон інверсії. Викривлення радіолокаційних сигналів також може впливати на точність визначення місця розташування об'єкта. Додатково підвищений рівень шуму в радарній системі створює перешкоди та аномалії у процесі виявлення.

2.3. Приклади впливу аномального поширення УКХ-радіохвиль на роботу АІС

Відповідно до рекомендацій ITU-R M.212, максимальна дальність зв'язку системи АІС визначається механізмами поширення радіохвиль у зоні

					171.6321M.KP.ПЗ.P2	Арк.
						38
Змн.	Арк.	№ документа	Підпис	Дата		

прямої видимості, а також явищами дифракції. При стандартних технічних параметрах бортового та берегового обладнання АІС найбільш надійна передача радіосигналів над морською поверхнею досягається на відстанях 20–35 морських миль, що залежить від висоти антен над рівнем моря.

Однак у певні періоди спостерігаються аномальні атмосферні умови, за яких зона дії АІС значно перевищує стандартні значення. Такі явища, як тропосферне розсіювання, формування атмосферних хвильоводів, дощове розсіювання, метеорні спалахи або спорадичний шар Е, можуть забезпечити передачу сигналів на значно більші відстані. Проте ці механізми не лише розширюють зону покриття, а й спричиняють ряд проблем. Зокрема, вони можуть викликати перевантаження цифрових ліній зв'язку УКХ-діапазону, що знижує здатність бортового обладнання АІС знаходити вільні часові інтервали для передачі сигналу. Це створює перешкоди в роботі системи, особливо в умовах високої щільності морського руху, і вимагає адаптації роботи АІС до таких явищ.

Перевантаження каналу зв'язку може суттєво порушити надійність ідентифікації суден та автоматичний обмін даними між суднами і береговими станціями. Зокрема, це ускладнює отримання ключової інформації, такої як координати, швидкість та курс суден. Унаслідок цього збільшується затримка у виявленні небезпечних маневрів суден, що значно підвищує ризик виникнення аварійних ситуацій. Описані вище явища аномального поширення радіохвиль, зокрема тропосферне розсіювання або атмосферні хвильоводи, можуть погіршити якість роботи автоматичної ідентифікаційної системи, що напряду впливає на безпеку морського судноплавства. Таким чином, дослідження механізмів поширення радіохвиль та їхнього впливу на роботу АІС є важливим аспектом забезпечення безпеки на морі.

Розглянемо наддальний прийом сигналів АІС [9]. Дослідження впливу аномального поширення УКВ-радіохвиль на роботу АІС проводилося на основі даних, отриманих зі станцій АІС № 4757 (Миколаїв) та № 5064

					171.6321M.KP.ПЗ.P2	Арк.
						39
Змн.	Арк.	№ документу	Підпис	Дата		

(Очаків), що належать компанії MarineTraffic і розміщені на базі Національного університету кораблебудування імені адмірала Макарова. З серпня 2020 року всі прийняті станціями повідомлення у форматі AIVDM систематично записувалися в базу даних. Для детального аналізу були обрані тимчасові інтервали, що характеризуються середньою та високою дальністю прийому сигналу. Кількість прийнятих AIVDM-пропозицій за обрані дні, наведена в таблиці 2.1, дозволила оцінити інтенсивність судноплавства та виявити можливі аномалії, пов'язані з особливостями поширення радіохвиль.

Таблиця 2.1 Статистика AIVDM-повідомлень

Кількість AIVDM-повідомлень		
Дата	Миколаїв	Очаків
2021-06-11	203 424	129 904
2021-06-12	193 779	333 031
2021-06-29	235 638	368 518
2021-06-30	255 406	511 095
2021-07-01	184 952	834 898
2021-07-02	159 127	323 102
Загалом	1 232 325	2 500 54

Аналіз даних, отриманих зі станцій AIS, виявив, що відмінності у кількості прийнятих повідомлень зумовлені комплексом факторів. По-перше, кількість суден у ближній зоні кожної станції суттєво впливає на кількість отриманих сигналів. По-друге, географічне розташування станцій: Миколаївська станція зазвичай фіксує більше суден, які знаходяться у постійній зоні її дії. По-третє, переважаюча морська поверхня та близькість Очакова до основних морських шляхів у Чорному морі сприяють збільшенню дальності прийому сигналів цією станцією.

					171.6321M.KP.ПЗ.P2	Арк.
						40
Змн.	Арк.	№ документа	Підпис	Дата		

Аналіз показав, що період з 11 по 12 червня відповідав нормальним умовам поширення радіохвиль. Однак, починаючи з 29 червня та досягнувши максимуму 1 липня, спостерігалось аномальне збільшення дальності прийому сигналів Очаківською станцією, що дозволило фіксувати судна навіть у районі Мармурового моря (рис. 2.5).

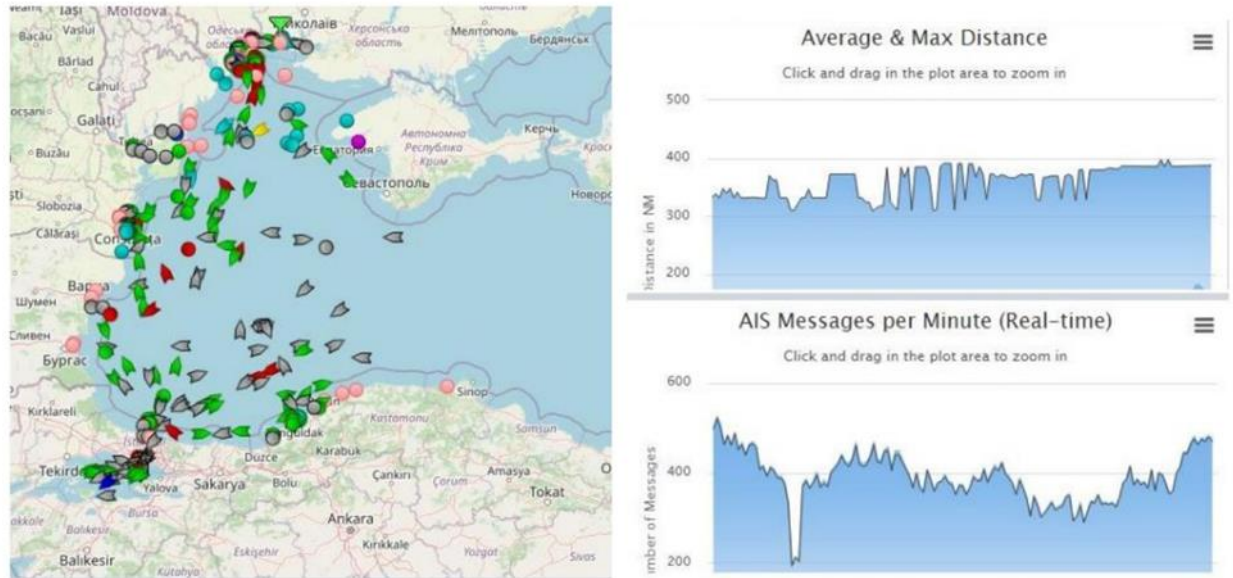


Рисунок 2.5 – Зона прийому сигналів AIS на станції Очаків
30 липня 2021 р.

Аналіз даних про прийом сигналів станції AIS в Очакові виявив тісний зв'язок із метеорологічними умовами. Карта індексу тропосфери Хепберна (НТІ) (рис. 2.6), яка характеризує потенційні шляхи поширення радіохвиль у тропосфері, продемонструвала високу кореляцію із зоною прийому сигналів станції. Високі значення НТІ свідчать про можливість утворення хвилеводів, що сприяють збільшенню дальності поширення радіохвиль.

Динаміка зміни сили сигналу також пов'язана з метеорологічними умовами. Наприклад, у нічний час і перед проходженням теплих фронтів спостерігаються нестабільності у середній і верхній атмосфері, що призводить до зміни інтенсивності сигналу. Незважаючи на те, що покращення умов поширення не мало значного впливу на загальну кількість

прийнятих сигналів Миколаївською станцією, зона її прийому суттєво розширилася (рис. 2.7).

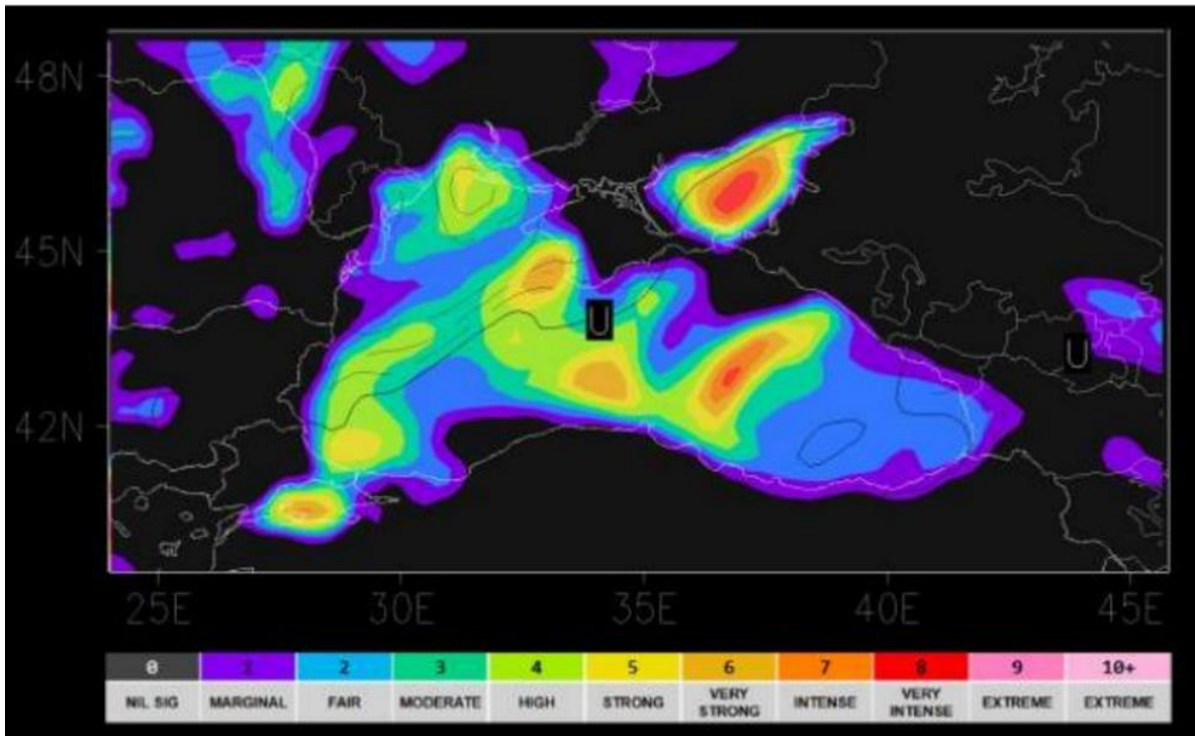


Рисунок 2.6 – Індекс тропосфери Хепберна, дійсний о 15:00 UTC+3 30 липня 2021 р., регіон Чорного моря (U – зони нестабільного сигналу)

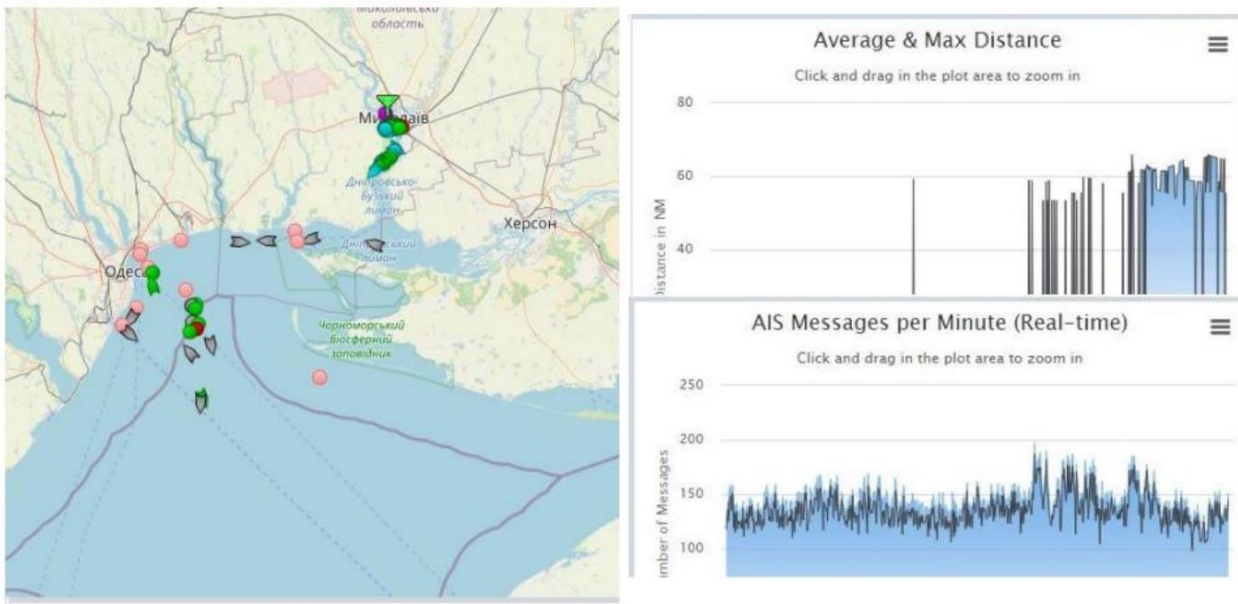


Рисунок 2.7 – Зона прийому сигналів AIS на станції Миколаїв 30 липня 2021 р.

Аномальне УКХ тропосферне поширення може призводити до зростання завантаження каналу обміну даними (VDL) і, як наслідок, до деградації якості морських радіозв'язків. Згідно з рекомендаціями [10], у випадку, якщо завантаження VDL перевищує 50% від номінальної кількості доступних тайм-слотів протягом останніх 4 хвилин на обох каналах, це може суттєво впливати на здатність мобільних станцій AIC знаходити вільні тайм-слоти для передачі даних.

Моніторинг завантаження VDL можна здійснювати за допомогою підрахунку кількості зайнятих тайм-слотів у кожному каналі відносно загальної кількості доступних тайм-слотів, яка становить 2250 на один кадр TDMA [6]. Отриманий показник зазвичай виражається у відсотках і може бути розрахований автоматизовано, якщо обладнання AIC оснащено спеціалізованими засобами для моніторингу каналу зв'язку.

Динамічні дані суден класу А передаються через позиційні звіти у повідомленнях №1, №2 та №3, відповідно до стандарту [12].

- Повідомлення №1 (Position Report 1): Це базове повідомлення, що передає найважливіші дані про судно, такі як його точні координати (широта та довгота), курс, швидкість і тип судна.
- Повідомлення №2 (Position Report 2): Це повідомлення містить додаткові дані, наприклад, статус судна (на якій швидкості воно рухається), а також інформацію про режим його роботи (чи рухається воно в автономному режимі чи отримує координати через GPS).
- Повідомлення №3 (Position Report 3): Воно містить розширену інформацію про рух судна, зокрема для великих суден або тих, що здійснюють важливі маневри, і забезпечує додаткові деталі щодо їх стану або вимог до безпеки.

Згідно з типовими параметрами, мобільні станції передають дані з середнім інтервалом $R_{Iavg} = 10$ секунд, чергуючи канали. При цьому, враховуючи доступність 2250 тайм-слотів на хвилину для кожного каналу,

					171.6321M.KP.ПЗ.P2	Арк.
						43
Змн.	Арк.	№ документа	Підпис	Дата		

критичний поріг завантаження VDL у 50% відповідає одночасній наявності близько 375 суден у зоні дії станції AIC радіусом 50 морських миль.

Таким чином, при досягненні цього порогу рекомендовано розпочинати детальний моніторинг каналу з боку компетентних органів для своєчасного виявлення потенційних проблем у роботі системи AIC.

$$VDL = N_t \frac{R_r}{2 \cdot 2250} \cdot 100\%,$$

де $R_r = 60/RI_{avg}$ – середня частота повідомлень, $N_t = N_v + N_{bs}$ – кількість цілей (N_v – кількість суден, N_{bs} – кількість базових станцій)

Згідно з стандартом [12], повідомлення AIC з позиційними звітами № 1, 2, а також повідомлення від базової станції № 4 містять параметри стану зв'язку SOTDMA. До цих параметрів належать: синхронізація сигналу, тайм-аут слота та додаткове підповідомлення. Коли значення параметра "тайм-аут слота" дорівнює трьом, п'яти або семи, підповідомлення стану зв'язку SOTDMA повинно містити інформацію про кількість прийнятих цілей або станцій.

Рисунок 2.8 демонструє кількість прийнятих станцій та оцінене завантаження VDL в процентному співвідношенні, яке обчислюється за формулою, для базових станцій системи управління рухом суден (VTS) за період з 2021-07-26 00:00:00 (UTC+3) до 2021-07-30 23:59:59 (UTC+3). У таблиці 2.2 наведено приклад для VTS з MMSI 2723682 (Очаків), де під час аномального поширення УКВ-хвиль завантаження VDL змінюється від мінімального значення 4,5% до максимального 89,6%. Пік завантаження VDL для цієї станції спостерігався 2021-07-29 о 19:45:00 UTC+3. Приклад карти завантаження VDL для цього моменту часу представлений на рис. 2.9.

					171.6321M.KP.ПЗ.P2	Арк.
						44
Змн.	Арк.	№ документа	Підпис	Дата		

Таблиця 2.2 – Макс. та мін. навантаження на VDL у системі VTS

MMSI	місце	Широта	Довгота	Завантаження VDL			
				Мін.		Макс.	
				Прийняті станції	%	Прийняті станції	%
2723666	Миколаїв	46.6310	31.8588	40	5,3	307	40,9
2723679	Південне	46.6011	31.0223	88	11,7	543	72,4
2723628	Курортне	30.2865	30.2865	31	4,1	420	56
2723627	Затока	46.0740	30.4696	26	3,7	349	46,5
2723682	Очаків	46.6121	31.5506	34	4,5	672	89,6
2723623	Очаків	46.6008	31.5501	67	8,9	642	85,6
2723703	Південне	46.6545	31.0337	70	9,3	570	76

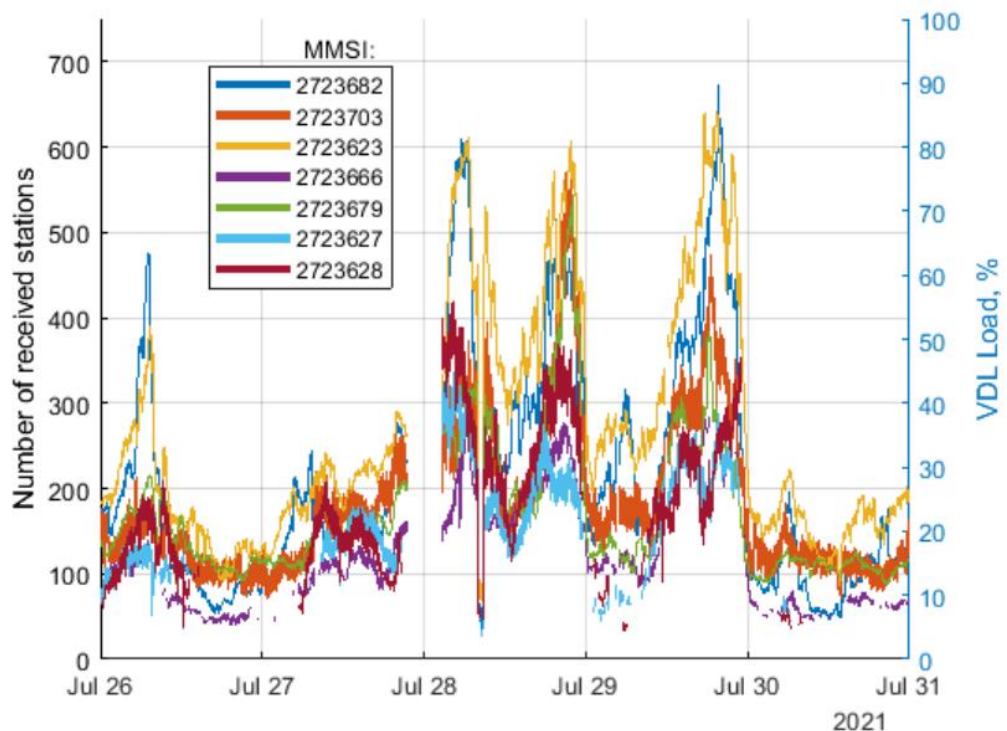


Рисунок 2.8 – Кількість прийнятих станцій для базових станцій системи управління рухом суден (VTS) за період з 26 липня 2021 року по 30 липня 2021 року включно (UTC+3)

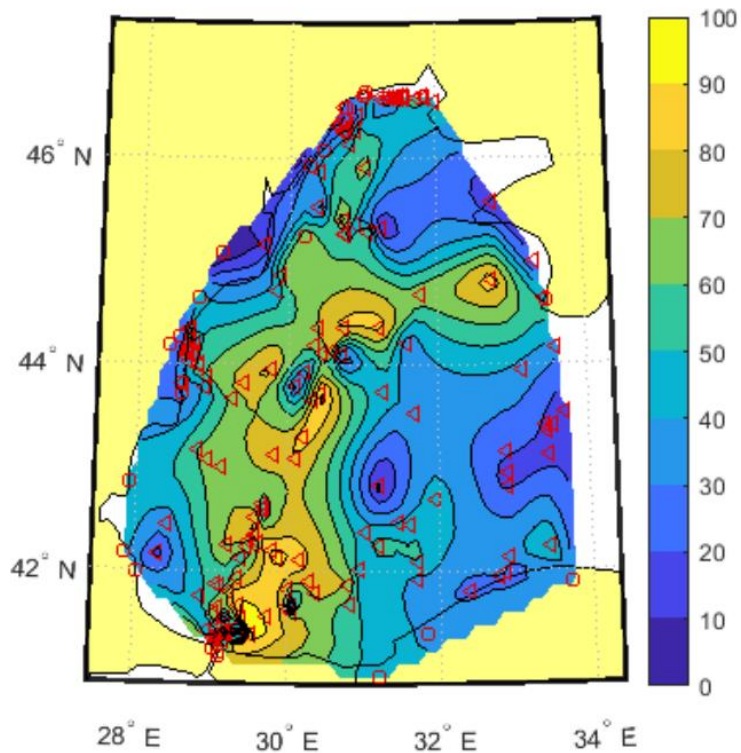


Рисунок 2.9 – Приклад карти навантаження VDL під час аномального розповсюдження УКВ-тропосферних хвиль 29 липня 2021 року о 19:45:00 UTC+3 (Δ – судна, О – базові станції)

2.4. Можливість прогнозування завантаженості VDL каналу AIS станцій на основі тропосферних карт

Одним із критичних аспектів функціонування AIS є передача даних через VHF Data Link (VDL) канал. Завантаженість цього каналу може суттєво змінюватися внаслідок впливу різних фізичних факторів, зокрема аномального тропосферного поширення радіохвиль. У статті [9] розглянуто зв'язок між цими явищами та їх вплив на площу покриття AIS станцій і завантаженість VDL каналу.

В нормальних умовах радіосигнали у цьому діапазоні поширюються майже по прямій лінії, злегка викривленій через рефракцію в тропосфері. Проте аномальні явища можуть значно змінювати траєкторію радіохвиль. Ці явища виникають через градієнти температури, вологості та тиску в нижніх шарах атмосфери, що впливає на діелектричні властивості повітря.

За умов аномального поширення сигналів радіохвилі поширюються на значно більшу відстань, ніж зазвичай. Це призводить до збільшення кількості суден, що передають дані через конкретну AIS станцію, і, як наслідок, до перевантаження VDL каналу.

Тропосферні карти, які складаються на основі метеорологічних даних, відображають просторовий розподіл ключових параметрів атмосфери: температури, тиску, вологості. Ці карти можна використовувати для прогнозування аномальних явищ у тропосфері, що впливають на поширення VHF сигналів. За їх допомогою можливо:

- Прогнозувати тропосферні умови: карти дозволяють ідентифікувати області, де очікуються інверсії чи утворення хвильових каналів, які впливають на поширення радіохвиль.
- Моделювати покриття AIS станцій: використовуючи тропосферні карти як вхідні дані для моделей поширення радіохвиль, можна створити прогнози площі покриття конкретної станції.
- Передбачати завантаженість VDL каналу: знаючи, як зміниться зона покриття, можна оцінити кількість суден у цій зоні та ймовірну завантаженість каналу.

Для вирішення цієї задачі можна запропонувати наступний алгоритм

- Збір метеорологічних даних: отримання тропосферних карт від метеорологічних служб або спеціалізованих систем спостереження.
- Оцінка зони покриття: визначення змін площі покриття AIS станції за різних тропосферних умов.
- Адаптація частотного ресурсу: у разі прогнозування перевантаження - оптимізація частотного ресурсу.

Використання тропосферних карт для прогнозування покриття AIS станцій є перспективним підходом, який дозволяє заздалегідь визначати потенційні проблеми із завантаженістю VDL каналу.

					171.6321M.KP.ПЗ.P2	Арк.
						47
Змн.	Арк.	№ документа	Підпис	Дата		

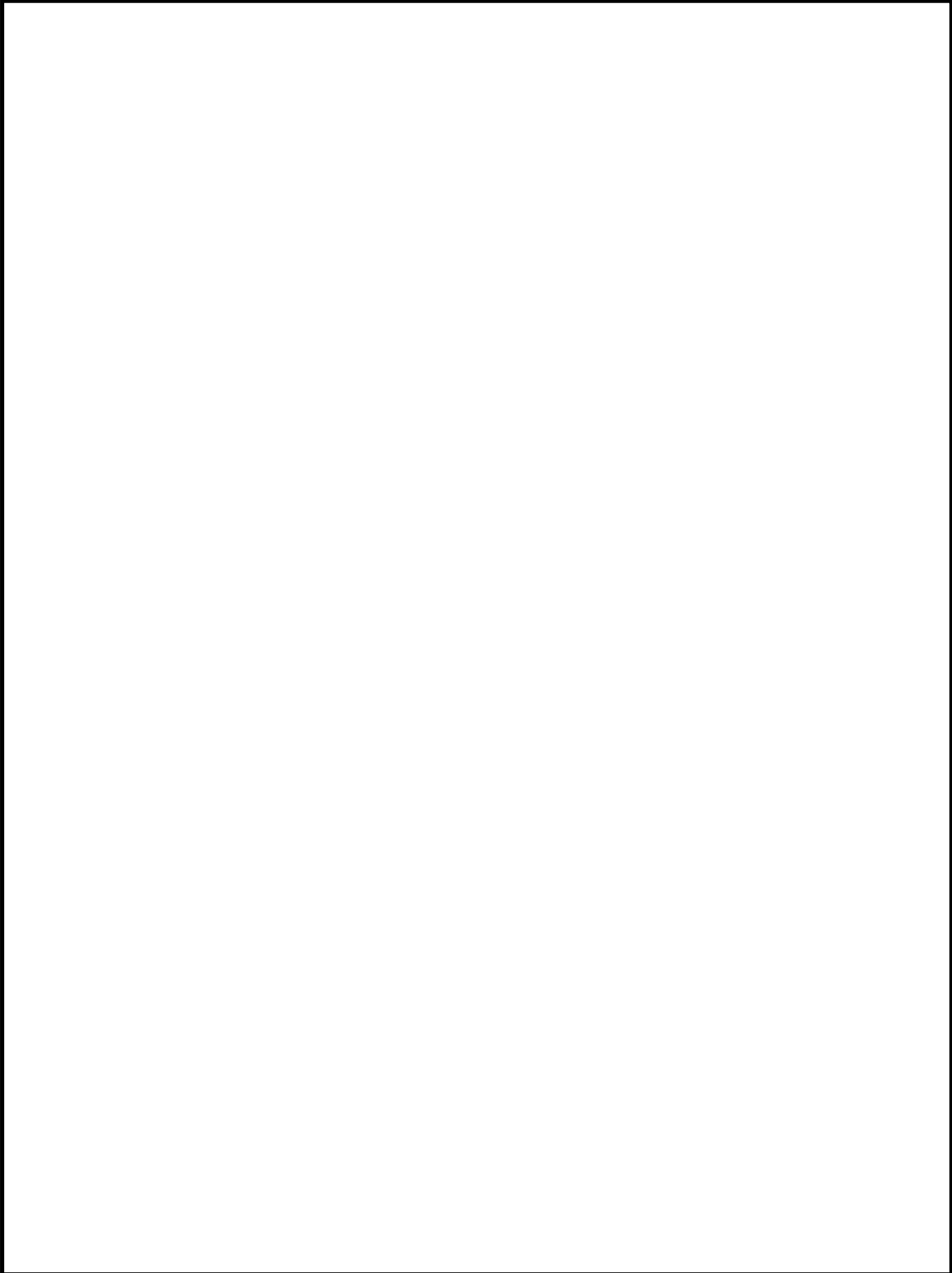
Висновок

У розділі було проведено детальний аналіз впливу аномальних профілів градієнтів рефрактивності на формування хвильоводів радіосигналів у діапазоні ультракоротких хвиль (УКХ). Зокрема, розглянуто такі типи хвильоводів, як піднятий хвильовід, поверхневі хвильоводи та випаровувальні хвильоводи. Ці явища виникають внаслідок специфічних атмосферних умов, зокрема інверсій температури, високої вологості та градієнтів тиску, що створюють аномальні умови для поширення радіохвиль.

Проаналізовано приклади впливу аномального поширення УКХ-радіохвиль на функціонування автоматичної ідентифікаційної системи (AIS). Зокрема, було визначено, що такі умови можуть призводити до значного збільшення зон покриття AIS станцій, що впливає на завантаженість VHF Data Link (VDL) каналу. Це може спричиняти перевантаження каналів через підвищення кількості суден які виходять за межі нормального покриття станції

Окремо розглянуто можливість використання тропосферних карт для прогнозування площі покриття AIS станцій і завантаженості VDL каналу.

					171.6321M.KP.ПЗ.P2	Арк.
						48
Змн.	Арк.	№ документу	Підпис	Дата		



					171.6321М.КР.ПЗ.РЗ			
Змн.	Арк.	№ документа	Підпис	Дата				
Розробник		Вишневський В. В.			Дослідження можливості прогнозування щавантаження АІС за погодними картами тропосферного індексу	Літ.	Арк.	Аркуші
Консультант							49	
Н. контр.		Добрінова Л. П.				НУК імені адмірала Макарова		
Керівник		Дьяконов О. С.						
Зав. каф.		Рябенський В. М.						

РОЗДІЛ 3. ДОСЛІДЖЕННЯ МОЖЛИВОСТІ ПРОГНОЗУВАННЯ ЗАВАНТАЖЕННЯ АІС ЗА ПОГОДНИМИ КАРТАМИ ТРОПОСФЕРНОГО ІНДЕКСУ

3.1. Вихідні дані та постановка задачі

Метою є розробка нейромережі для прогнозування області покриття АІС станції на основі карт індексу тропосфери Хепберна (Hepburn Tropo Index, HTI). Необхідно навчити модель прогнозувати покриття станції, використовуючи карти HTI як вхідні дані.

Аналіз сигналів прийому станції АІС в Очакові [9] виявив сильну кореляцію між метеорологічними умовами та зоною покриття станції. Карти HTI, що відображають потенційні шляхи поширення радіохвиль у тропосфері, демонструють зв'язок із формуванням хвилеводів, які сприяють розширенню зони покриття. Розробка моделі прогнозування зон покриття допоможе оптимізувати використання станцій для зв'язку.

Етапи розв'язання задачі:

- Збір і підготовка даних.
- Вибір моделі.
- Розробка моделі.
- Навчання і тестування моделі.
- Оцінка продуктивності.

Очікуваний результат: створення моделі, яка здатна з високою точністю відтворювати зони покриття станції на основі карт HTI, що сприятиме оптимізації роботи радіозв'язкових станцій.

В якості вихідних даних виступають карти індексу тропосфери Хепберна на сайті, можна отримати прогноз з 3-годинними інтервалами на перші 36 годин і з 6-годинними інтервалами на 6 днів, що залишилися. Карти оновлюються щодня близько 18:00 UTC. Ці прогнози подаються у вигляді PNG-зображень, і доступні для різних регіонів планети, в випадку данної роботи використовуються прогнози для Східної Європи.

					171.6321M.KP.P3.P3	Арк.
						50
Змн.	Арк.	№ документу	Підпис	Дата		

Також для розробки необхідні дані про область покриття АІС-станції; ці дані відображаються на сайті MarineTraffic, де вони відображені у вигляді мап покриття (рис. 3.1). Зазначимо, що напряду використовувати таке візуальне представлення даних не вийде.

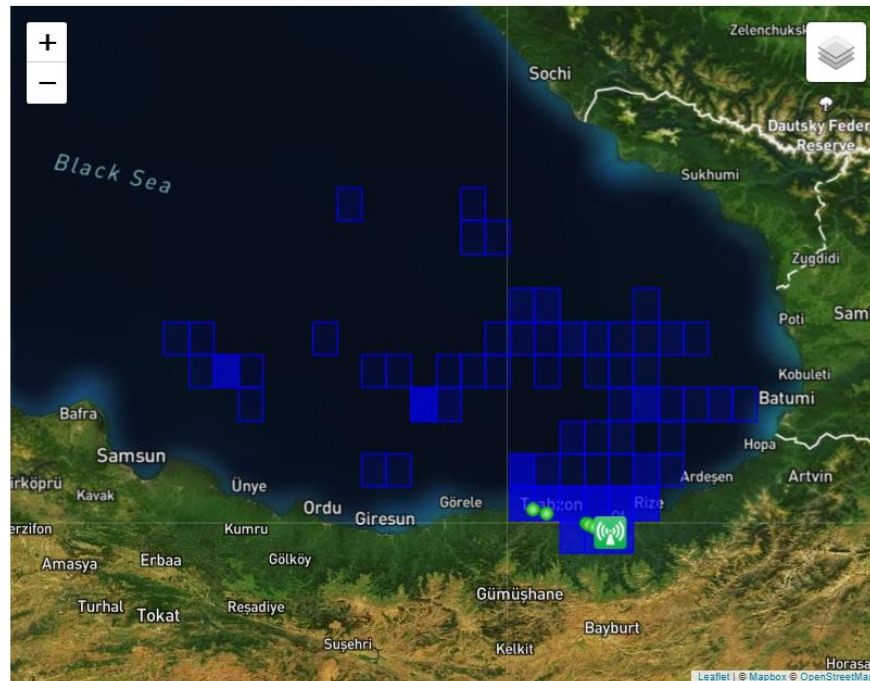


Рисунок 3.1 – Приклад відображення області покриття станції АІС на сайті MarineTraffic

Проте, якщо уважно вивчити код сторінки, використовуючи консоль розробника, можна помітити XHR-запит – об'єкт JavaScript, який використовується для асинхронного обміну даними між браузером і сервером (рис 3.2). Цей запит повертає відповідь від сервера, json файл, на основі якого сайт і відображає мапу покриття. Цей json містить в собі координати (lat, lon), які можна використовувати для побудови власної мапи покриття.

Тобто, для роботи необхідні дані з двох сайтів, які необхідно отримувати за однакові часові проміжки. Для автоматизації цього процесу можна скористатись скрапінгом.

					171.6321М.КР.ПЗ.РЗ	Арк.
						51
Змн.	Арк.	№ документа	Підпис	Дата		

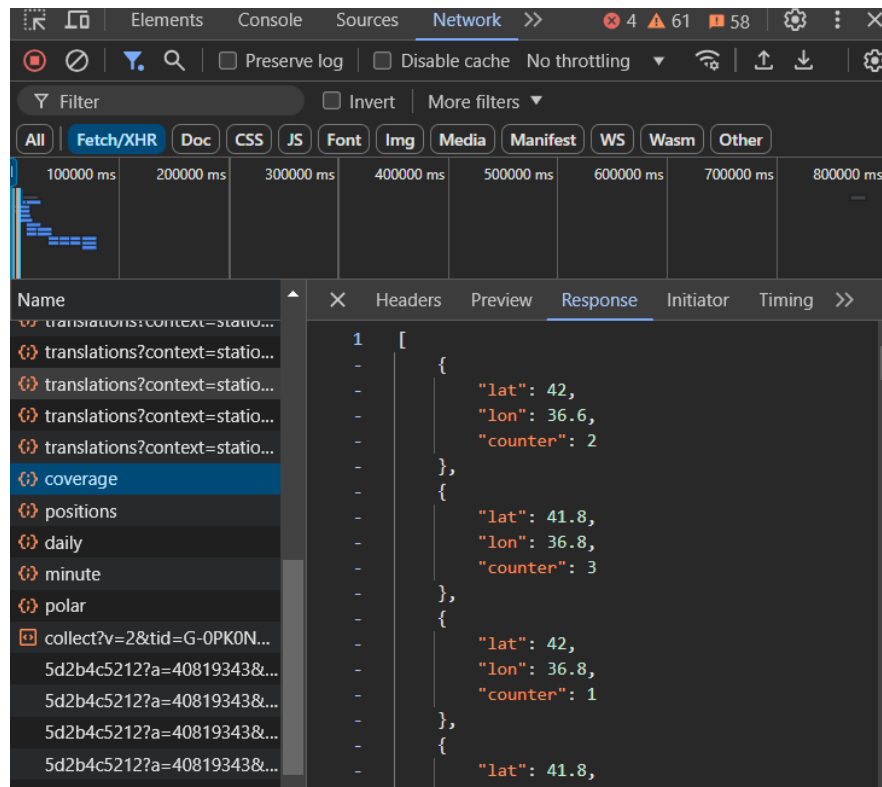


Рисунок 3.2 – приклад json файлу який сервер повертає у якості відповіді(response) на XHR запит

Скрапінг (від англ. web scraping) - це автоматизований процес збору даних із веб-сайтів за допомогою спеціальних програм або скриптів. Скрапінг використовується для отримання потрібної інформації з веб-сторінок, яку потім можна зберігати та аналізувати.

Принцип роботи:

- Скрипт або програма отримує HTML-код веб-сторінки.
- Дані зі сторінки витягуються на основі певних шаблонів або елементів (теги, класи CSS, атрибути).
- Зібрану інформацію можна зберегти в базу даних, файл (CSV, JSON тощо) або використовувати для подальшої обробки.

Для скрапінгу можна використовувати мову програмування Python, для подібних задач реалізовано велику кількість бібліотек. Деякі з них використовуються в роботі.

Requests [13] – це потужна бібліотека Python, спеціально розроблена для спрощення взаємодії з веб-серверами. Вона дозволяє надсилати різноманітні HTTP-запити (GET, POST, PUT, DELETE тощо) та отримувати відповіді від них. Основні функціональні можливості:

- Відправка GET-запитів: за допомогою методу `requests.get()` можна отримати дані з вказаного URL.
- Відправка POST-запитів: метод `requests.post()` дозволяє відправляти дані на сервер.
- Налаштування заголовків: за допомогою параметра `headers` можна задавати додаткові заголовки для запиту.
- Передача даних у тілі запиту: для передачі даних у тілі запиту використовується параметр `data` або `json`.
- Обробка відповідей: отриману відповідь можна обробити за допомогою атрибутів `status_code`, `headers`, `text` та інших.

Для автоматизованого отримання карт прогнозів тропосферного розповсюдження було реалізовано скрипт, який приведений в додатку А; цей код автоматично завантажує зображення з сайту та організовує дані у вигляді папок ,структурованих за датами (рис. 3.3 а).

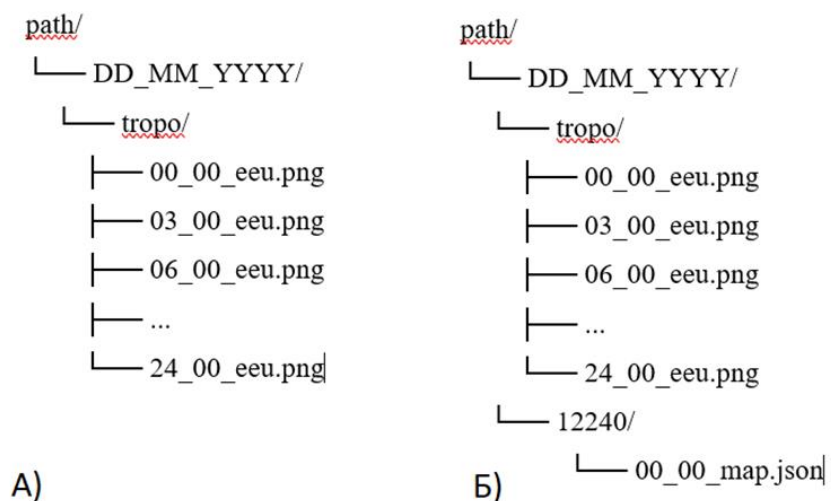


Рисунок 3.2 – а) приклад організації мап НТІ, б) приклад організації інформації про покриття з Marine Traffic

Також було автоматизовано отримання json-файлів з Marine Traffic. Для цього було написано скрипт, який приведено у додатку. Він використовує бібліотеку selenium.

Selenium [14] – це набір інструментів для автоматизації веб-браузерів. Основна мета Selenium - автоматизувати взаємодію користувача з веб-сторінками, включаючи кліки, введення тексту, навігацію та виконання JavaScript. Основні можливості Selenium:

- Знаходження елементів за ID, класом, CSS-селекторами, XPath тощо.
- Клік, введення тексту, прокрутка сторінки.
- Навігація між сторінками:
- Відкриття URL: `driver.get("url")`.
- Перехід назад/вперед: `driver.back()`, `driver.forward()`.
- Отримання тексту: `element.text`.
- Отримання атрибутів: `element.get_attribute("attribute_name")`.
- Відкриття нових вкладок: `driver.execute_script("window.open('url')")`.
- Переключення між вкладками: `driver.switch_to.window(handle)`.
- Запуск JavaScript:

Скрипт скрапить сайт та збирає json-файли організовуючи їх у папки, як показано на рисунку 3.3, б. Скрапінг цих сайтів допомагає автоматично отримати вихідні дані для тренування моделі.

3.2. Обробка та підготовка даних

Дані отримані при скрапінгу не вийде одразу використовувати для навчання моделі, спочатку їх необхідно підготувати під той вигляд, який необхідний нейромережі. Приблизна схема обробки даних приведена на рисунку 3.4. Для початку необхідно вилучити інформацію з PNG-зображення мапи тропосферного індексу. Для цього було написано скрипт `Create_geotiff`, він приведений в додатку Б

					171.6321M.KP.ПЗ.РЗ	Арк.
						54
Змн.	Арк.	№ документа	Підпис	Дата		

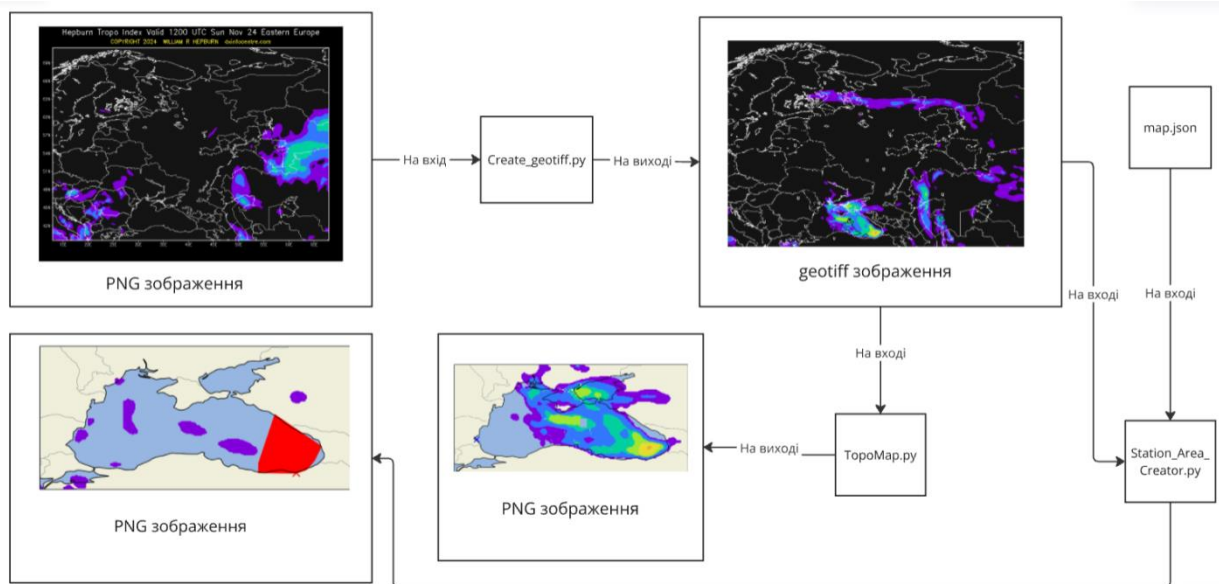


Рисунок 3.4 – Схема підготовки даних для навчання нейромережі

Першим кроком необхідно визначити ті пікселі на зображенні тропосферної карти, координати яких нам відомі (рис 3.5).

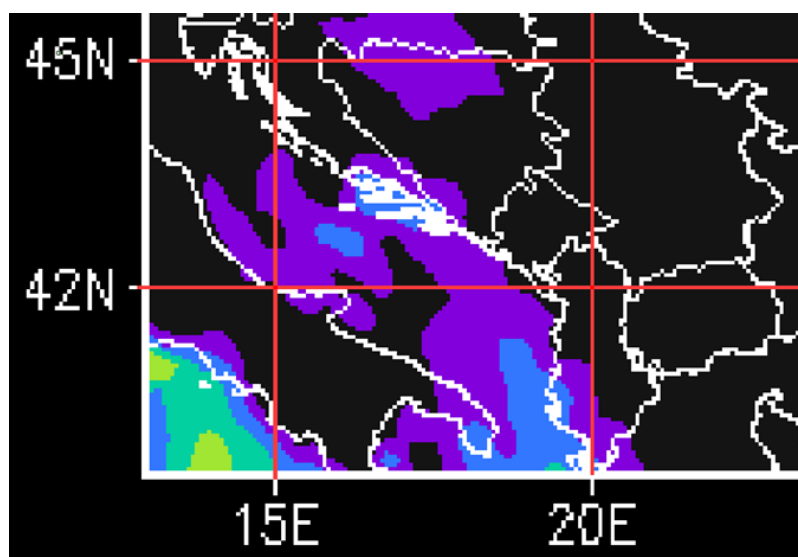


Рисунок 3.5 – Демонстрація пошуку координат відомих пікселів

Візуально на перетині червоних ліній можна побачити піксель та по координатній сітці зрозуміти їх координати. Використавши всі відомі пікселі можна вирахувати координати пікселів на краях зображення. Маючи цю інформацію стає можливим зробити геореференцію зображення.

Геореференція - це процес прив'язки просторових даних (наприклад, зображень, карт або будь-яких інших географічних даних) до реальної географічної системи координат. Завдяки цьому об'єкти на карті чи зображенні можуть бути зіставлені з конкретними місцями на земній поверхні.

Принцип геореференції:

- Вихідні дані: наприклад, растрове зображення (карта, супутниковий знімок тощо), яке спочатку не має географічних координат.
- Визначення контрольних точок: знаходяться точки на зображенні, координати яких відомі в реальному світі.
- Прив'язка: за допомогою математичних трансформацій зображення коригується таким чином, щоб його пікселі відповідали реальним координатам.
- Збереження: результат зберігається в форматі, який підтримує геоприв'язку, наприклад GeoTIFF, або як набір метаданих.

Скрипт `Create_geotiff` перебирає директорії (рис 3.3, б) знаходить всі png-файли з картами. Обрізає зображення, щоб позбутись рамки, яка буде заважати при подальшій обробці. Основна функція `process_image`: відкриває PNG-зображення у кольоровому форматі (RGB). Перетворює обрізане зображення у NumPy-масив. Після чого створює GeoTIFF-файл із: трьома каналами (червоний, зелений, синій), географічною системою координат (широта/довгота), та Affine-трансформацією (побудова зв'язку між піксельними координатами та географічними межами). Для цього було використано бібліотеку `rasterio`.

Rasterio [15] – це потужна бібліотека Python, спеціально розроблена для роботи з растровими геопросторовими даними. Вона забезпечує простий та ефективний спосіб читання, запису та обробки різноманітних форматів растрових зображень, які широко використовуються в геоінформаційних системах (ГІС).

					171.6321М.КР.ПЗ.РЗ	Арк.
						56
Змн.	Арк.	№ документа	Підпис	Дата		

Основні можливості Rasterio:

- Читання та запис растрових даних: простий інтерфейс для відкриття та збереження растрових файлів у різних форматах.
- Доступ до метаданих: отримання інформації про растровий файл, такої як розмір пікселя, система координат, число каналів.
- Маніпуляції з растровими даними: обрізання, масштабування, перепроєктування, обчислення статистики та інші операції з растровими даними.
- Використання масок: виділення областей інтересу на растровому зображенні за допомогою масок.
- Інтеграція з іншими бібліотеками: легко інтегрується з іншими бібліотеками Python, такими як Shapely, Pandas та Matplotlib.

Після отримання geotiff файлів наступним за схемою (рис.3.4) кроком буде створення нових мап. Мапи з тропосферними даними та мапи з покриттям станції.

Для початку необхідно отримати дані зі створеного geotiff-файла. На мапі тропосферний індекс позначено лінійною шкалою від 1 до 10, тому цю шкалу легко представити як масив (рис.3.6).

```
target_color = {
    'Marginal': numpy.array([130, 0, 220]),
    'Fair': numpy.array([51, 119, 255]),
    'Moderate': numpy.array([2, 208, 161]),
    'High': numpy.array([160, 230, 50]),
    'Strong': numpy.array([230, 220, 50]),
    'Very_Strong': numpy.array([230, 175, 45]),
    'Intense': numpy.array([240, 130, 40]),
    'Very_intense': numpy.array([250, 60, 60]),
    'Extreme': numpy.array([255, 128, 192]),
    'Extreme+': numpy.array([255, 180, 220])
}
```

Рисунок 3.6 – Масив з тропосферними індексами прив'язаними до кольорів

В спеціальному скрипті, який наведено в додатку В, ми можемо відкрити geotiff-файл та скористатись функцією `get_unique_colors`(рис. 3.7).

					171.6321M.KP.ПЗ.РЗ	Арк.
						57
Змн.	Арк.	№ документу	Підпис	Дата		

```
def get_unique_colors(file_path): 1 usage

    with rasterio.open(file_path) as dataset:
        red = dataset.read(1)
        green = dataset.read(2)
        blue = dataset.read(3)
        colors = np.stack( arrays: (red, green, blue), axis=-1)
        unique_colors = np.unique(colors.reshape(-1, 3), axis=0)

    return unique_colors
```

Рисунок 3.7 – Функція для пошуку унікальних кольорів в geotiff

Ця функція читає зображення з GeoTIFF-файлу, після чого витягує значення пікселів для кожного каналу (R, G, B). Створює unique_colors – список унікальних кольорів у зображенні, після чого повертає його в форматі (R, G, B). Ця функція потім використовується у функції find_matching_colors_and_cords(рис .3.8).

```
def find_matching_colors_and_coords(file_path, target_colors=target_color): 18 usages
    unique_colors = get_unique_colors(file_path)
    matching_coords = {}
    with rasterio.open(file_path) as src:
        for label, target_color in target_colors.items():
            coords = get_pixel_coords_by_color(file_path, target_color)
            matching_coords[label] = coords
    return matching_coords
```

Рисунок 3.8 – Функція для отримання координат пік селів, що несуть в собі інформацію про тропосферне покриття

Ця функція отримує список усіх унікальних кольорів у GeoTIFF. Проходить по кожному заданому кольору (target_colors) та перевіряє його наявність серед унікальних кольорів. Для кожного знайденого кольору отримуються координати, після чого функція повертає словник де ключі - назви кольорів, а значення - масиви географічних координат.

					171.6321M.KP.ПЗ.РЗ	Арк.
						58
Змн.	Арк.	№ документа	Підпис	Дата		

Для побудови власної мапи використаємо бібліотеку Eomaps.

EOMaps [16] – це сучасна бібліотека Python, спеціально розроблена для ефективної візуалізації та аналізу геопросторових даних. Вона надає інструментарій для створення інтерактивних карт, що дозволяють не лише відображати географічну інформацію, але й проводити глибокий аналіз даних у контексті їх просторового розташування. Бібліотека пропонує простий та інтуїтивний синтаксис, що дозволяє швидко створювати складні геопросторові візуалізації.

Ключові особливості та можливості EOMaps:

- Підтримка різноманітних форматів даних: EOMaps працює з широким спектром геопросторових форматів, включаючи растрові дані (наприклад, супутникові зображення), векторні дані (наприклад, форми геометрій) та табличні дані, що містять географічні координати.
- Інтерактивність: карти, створені за допомогою EOMaps, є динамічними та інтерактивними. Користувачі можуть збільшувати масштаб, зменшувати масштаб, переміщати карту, вибирати певні об'єкти та отримувати детальну інформацію про них.
- Глибока інтеграція з екосистемою Python: EOMaps тісно інтегрується з іншими популярними бібліотеками Python, такими як NumPy, Pandas, GeoPandas та Хаgгау, що дозволяє ефективно обробляти та аналізувати геопросторові дані.
- Висока продуктивність: бібліотека оптимізована для роботи з великими обсягами даних, що робить її ідеальним інструментом для аналізу даних дистанційного зондування та інших великомасштабних геопросторових проєктів.
- Кастомізація: EOMaps надає широкий спектр інструментів для кастомізації карт, включаючи налаштування стилів, легенд, масштабних лінійок, а також додавання різних типів маркерів та підписей.

					171.6321М.КР.ПЗ.РЗ	Арк.
						59
Змн.	Арк.	№ документа	Підпис	Дата		

- Вихідні формати: створені карти можуть бути експортовані в різні формати, такі як, HTML, PNG, SVG та інші, що дозволяє легко інтегрувати їх в веб-додатки, звіти та презентації.

В додатку Г приведено повний код скрипта ТгороМар.ру (рис.3.4) для створення власної мапи. Розглянемо код створення та ініціалізації мапи (рис. 3.9).

```
m = Maps(facecolor="none", figsize=(6, 3.5))
m.add_feature.preset.coastline()
m.add_feature.preset.land()
m.add_feature.preset.ocean()
m.add_feature.preset.countries()
m.ax.set_extent(Map_size)
m.ax.scatter(*station_coords, color='blue', marker='x', s=100, label='Station LOT')
```

Рисунок 3.9 – приклад коду для створення мапи за допомогою бібліотеки EОmaps

Maps: це основний об'єкт класу з бібліотеки eomaps, який використовується для створення мапи. Він дозволяє додавати на мапу шари (наприклад, узбережжя, кордони країн тощо) і виконувати різні маніпуляції з географічними даними. facecolor="none": Цей параметр задає колір фону карти. "none" означає, що фон буде прозорим. Якщо необхідно задати колір, можна використати назву кольору або його шістнадцятковий код (наприклад, "white" або "#FFFFFF").

figsize=(6, 3.5): визначає розмір карти, розміри задають пропорції карти і впливають на її відображення при збереженні в файл.

m.add_feature.preset.coastline(): додає лінії узбережжя до карти, це важливий шар для ідентифікації контурів суші та води.

m.add_feature.preset.land(): додає заповнення для суші (землі). Типовий колір землі може залежати від налаштувань бібліотеки, але його можна змінити.

					171.6321M.KP.ПЗ.РЗ	Арк.
						60
Змн.	Арк.	№ документа	Підпис	Дата		

`m.add_feature.preset.ocean()`: додає заповнення для океанів чи морів, колір води також можна змінювати.

`m.add_feature.preset.countries()`: додає кордони країн на карту, це корисно для візуалізації адміністративних меж.

`m.ax.set_extent(Map_size)`: встановлює видиму область карти. `Map_size` - це список із чотирьох значень, які визначають межі карти в випадку даної роботи довгота від 26.06 до 43.18, широта від 40.02 до 47.92. що відповідає приблизним межах Чорного моря.

Загалом код `TropoMap.py` перебирає всі `geotiff`-файли, отримує з них координати пікселів з інформацією про тропосферні індекси та відображає їх на новоствореній мапі, після чого автоматично створює папку «Input» куди зберігає ці мапи (рис.3.10); вони будуть використовуватись як вхідне зображення при навчанні нейромережі

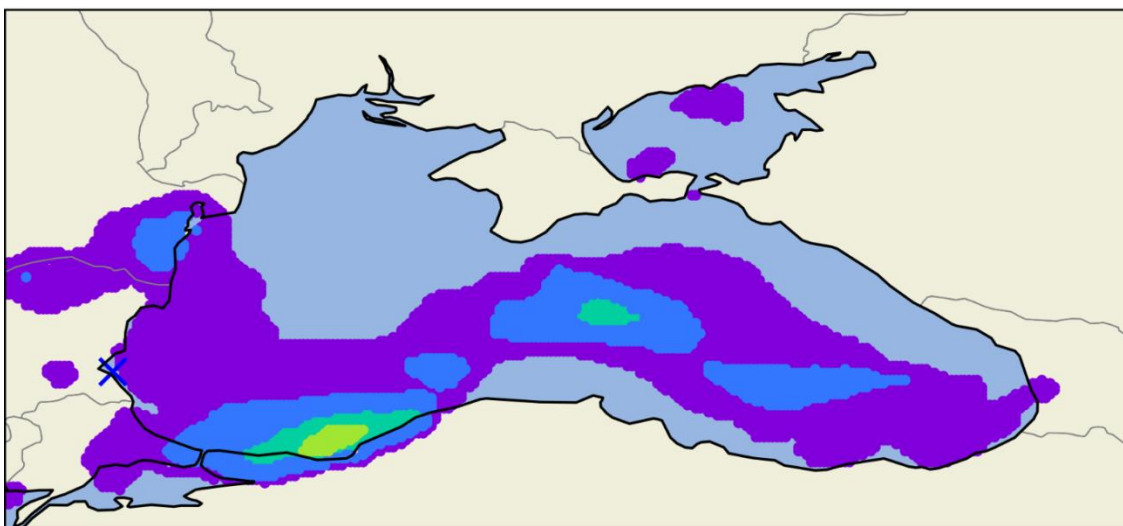


Рисунок 3.10 – мапа створена скриптом `TropoMap.py`

Далі за схемою (рис 3.4) розглянемо `Station_Area_creator.py`. Повний код приведено у додатку Г. Цей скрипт працює дуже подібно до попереднього, він так само малює мапу, проте додатково він аналізує `map.json`, отриманий з сайту `Marine Traffic`, та наносить на мапу полігон, який демонструє площу покриття.

```

with open(json_path, 'r') as f:
    coordinates = json.load(f)
    polygon_coords = [(float(item["LON"]), float(item["LAT"])) for item in coordinates]
    points = np.array(polygon_coords)
    hull = ConvexHull(points)
    hull_points = points[hull.vertices]
    polygon = Polygon(hull_points, closed=True, edgecolor='red', facecolor='red', fill=True, linewidth=2, zorder=10)
    m.ax.add_patch(polygon)
    m.ax.scatter(*station_coords, color='red', marker='x', s=100, label='Station L0T', zorder=11)
    m.ax.set_extent(Map_size)

```

Рисунок 3.11 – Частина скрипта для відображення покриття станції на мапі

JSON-файл відкривається та зчитується (рис 3.11). Передбачається, що у ньому міститься масив об'єктів з ключами "LON" (довгота) та "LAT" (широта), які представляють координати полігону. Зчитані координати перетворюються у список кортежів [(LON, LAT), ...], значення явно конвертуються у тип float для подальшої роботи. Після чого координати полігону конвертуються у формат numpy.array, а потім з них обчислюється опукла оболонка за допомогою функції ConvexHull з бібліотеки scipy. Опукла оболонка - це найменший багатокутник, який повністю охоплює всі точки. Координати вершин опуклої оболонки (hull.vertices) використовуються для побудови полігону. Полігон додається на карту (m.ax) як заповнений червоним кольором багатокутник з червоним контуром(рис 3.12).

					171.6321M.KP.ПЗ.РЗ	Арк.
						62
Змн.	Арк.	№ документа	Підпис	Дата		

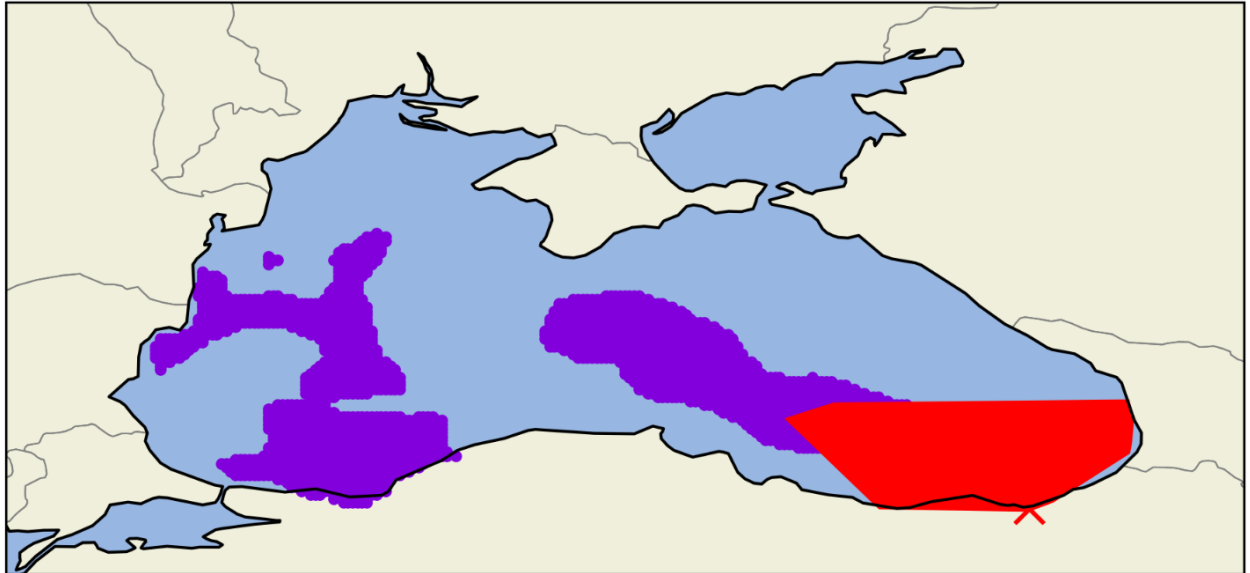


Рисунок 3.12 – Візуалізація покриття AIS-станції

Ці скрипти автоматично обробляють дані, створюючи і зберігаючи в відповідних папках зображення мап. В результаті отримано 226 зображень, зібраних за 44 дні (рис 3.13).



Рисунок 3.13 – Підготовлені дані

Останнім кроком буде створення бінарних масок на основі мап з візуалізацією покриття станцій. Код приведено в додатку Д. Для цього використовується бібліотека OpenCV.

					171.6321М.КР.ПЗ.РЗ	Арк.
						63
Змн.	Арк.	№ документа	Підпис	Дата		

OpenCV (Open Source Computer Vision Library) [17] – це одна з найпопулярніших і широко використовуваних бібліотек з відкритим кодом, призначена для розробки додатків у галузі комп'ютерного зору. Вона пропонує широкий спектр алгоритмів і функцій, які дозволяють обробляти зображення, відео та аналізувати візуальну інформацію з метою виявлення об'єктів, розпізнавання образів, відстеження руху та виконання багатьох інших завдань.

Основні можливості OpenCV:

- Зміна розміру, обертання, масштабування зображень.
- Конвертація між різними колірними просторами (RGB, HSV, grayscale тощо).
- Фільтрація зображень (згладжування, шумозаглушення, виділення країв).
- Морфологічні операції (ерозія, дилатація, відкриття, закриття).
- Виявлення ліній, кутів (детектори Харіса, Ши-Томасі), контурів.
- Опис текстур (LBP, HOG).
- Оцінка спотворень зображення.
- Стереовізійні алгоритми для визначення глибини.
- Вбудовані алгоритми машинного навчання для класифікації, регресії та кластеризації.
- Можливість інтеграції з зовнішніми бібліотеками машинного навчання (TensorFlow, PyTorch).

Спочатку зображення відкривається за допомогою OpenCV, після чого переводиться з формату RGB (BGR в OpenCV) у формат HSV (Hue, Saturation, Value). Це робиться для зручного виділення кольорів (у цьому випадку червоного) (рис. 3.14).

					171.6321М.КР.ПЗ.РЗ	Арк.
						64
Змн.	Арк.	№ документу	Підпис	Дата		

```

polygon_image = cv2.imread(file_path)
hsv = cv2.cvtColor(polygon_image, cv2.COLOR_BGR2HSV)
lower_red1 = np.array([0, 70, 50])
upper_red1 = np.array([10, 255, 255])
lower_red2 = np.array([170, 70, 50])
upper_red2 = np.array([180, 255, 255])
mask1 = cv2.inRange(hsv, lower_red1, upper_red1)
mask2 = cv2.inRange(hsv, lower_red2, upper_red2)
mask = mask1 | mask2
binary_mask = cv2.threshold(mask, 127, 255, cv2.THRESH_BINARY)[1]
cv2.imwrite(file_path, binary_mask)

```

Рисунок 3.14 – Частина коду для створення бінарних масок

Червоний колір у HSV-просторі має два діапазони: `lower_red1` та `upper_red1`: відтінки червоного ближче до 0 градусів. `lower_red2` та `upper_red2`: відтінки червоного ближче до 180 градусів.

`cv2.inRange` створює маску, де пікселі, що потрапляють у заданий діапазон, позначаються білим (255), а всі інші — чорним (0). Об'єднання масок (`mask = mask1 | mask2`) дозволяє врахувати обидва діапазони червоного кольору.

`binary_mask = cv2.threshold(mask, 127, 255, cv2.THRESH_BINARY)`
 Перетворю маску на бінарне зображення, де значення пікселів стають або 0 (чорний), або 255 (білий). Цей крок забезпечує чітке виділення областей. Після чого отримана бінарна маска зберігається (рис 3.15).

Отримані бінарні маски а також відповідні їм погодні мапи будуть використані для навчання нейромережі

					171.6321М.КР.ПЗ.РЗ	Арк.
						65
Змн.	Арк.	№ документа	Підпис	Дата		

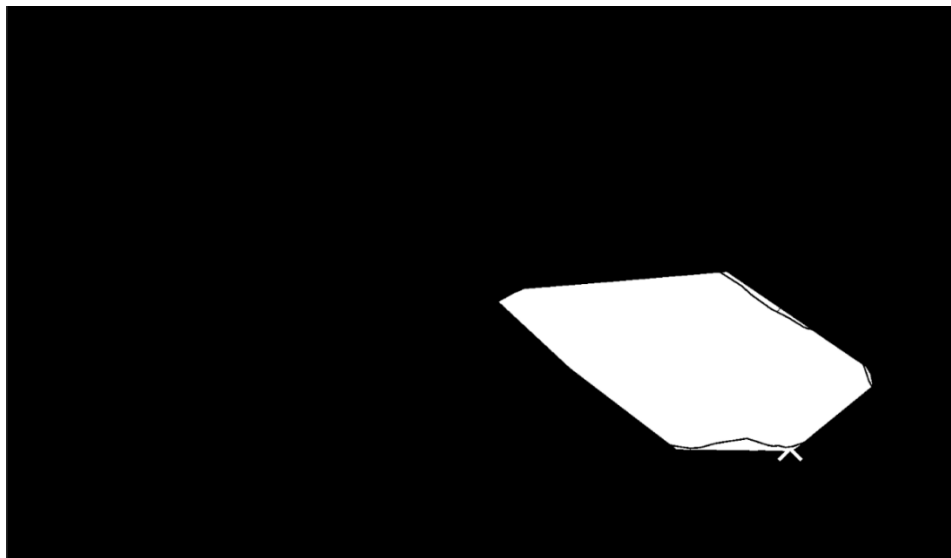


Рисунок 3.15 – Приклад отриманої бінарної маски

3.3. Вибір моделі. Навчання нейромережі

Нейромережа повинна прогнозувати полігон покриття по тропосферним картам. Для навчання мережі можна використати безкоштовний онлайн сервіс від Google (Google Colab), який дозволяє запускати код Python прямо у браузері. Це дозволяє не встановлювати на власний комп'ютер додаткове програмне забезпечення. Також Google Colab надає безкоштовний доступ до потужних обчислювальних ресурсів, включаючи графічні процесори (GPU) та тензорні процесори (TPU), що дозволяє ефективно тренувати навіть великі моделі машинного навчання.

Для створення нейромережі використовується бібліотека **TensorFlow** [18] - це популярна платформа для машинного навчання з відкритим вихідним кодом, розроблена компанією Google Brain. Вона використовується для створення, навчання та розгортання моделей машинного навчання, включаючи глибокі нейронні мережі.

Основні характеристики TensorFlow:

- TensorFlow підтримує як низькорівневе програмування, дозволяючи повний контроль над обчисленнями, так і високорівневі API (наприклад, Keras) для швидкої побудови моделей.

- TensorFlow працює на різних платформах, включаючи настільні комп'ютери, сервери, мобільні пристрої та вбудовані системи.
- Підтримує навчання моделей на одному комп'ютері або розподілене навчання на багатьох вузлах, що дозволяє обробляти великі обсяги даних.
- TensorFlow оптимізований для виконання обчислень на графічних процесорах (GPU) та тензорних процесорах (TPU), що значно прискорює навчання моделей.
- TensorFlow використовує обчислювальні графи для представлення та оптимізації моделі. Ця структура дозволяє легко відслідковувати залежності між операціями та автоматично обчислювати градієнти.
- TensorFlow написаний на Python і C++, але також має підтримку інших мов, таких як JavaScript (TensorFlow.js), Java, C#, та ін.

Keras - це високорівнева бібліотека для розробки та навчання нейронних мереж, яка є частиною екосистеми TensorFlow. Вона була створена з метою спрощення побудови моделей машинного навчання, що дозволяє дослідникам і розробникам швидко створювати прототипи нейронних мереж без необхідності заглиблюватися в технічні деталі низького рівня. Бібліотека надає широкий спектр вбудованих модулів для обробки зображень, тексту, серійних даних.

Основні особливості Keras:

- Keras дозволяє легко комбінувати різні компоненти (шари, функції активації, оптимізатори, функції втрат тощо) для створення складних моделей.
- З версії TensorFlow 2.0 Keras є його офіційною високорівневою API.
- Підтримка обчислення на GPU, що значно прискорює навчання моделей.
- Бібліотека keras.applications надає доступ до багатьох популярних архітектур (наприклад, ResNet, Inception, MobileNet), які можна

					171.6321M.KP.ПЗ.РЗ	Арк.
						67
Змн.	Арк.	№ документа	Підпис	Дата		

використовувати для донавчання або як основу для створення нових моделей.

- Keras підтримує додавання власних компонентів, таких як користувацькі шари, метрики та функції втрат.

Згорткова нейронна мережа (Convolutional Neural Network, CNN) - це нейронної мережі, спеціально розроблений для роботи з даними, що мають сітчасту структуру, наприклад, зображеннями або часовими рядами. CNN використовуються переважно для задач комп'ютерного зору, таких як класифікація зображень, розпізнавання об'єктів, сегментація зображень і багато іншого основна архітектура CNN приведена на рис. 3.16.

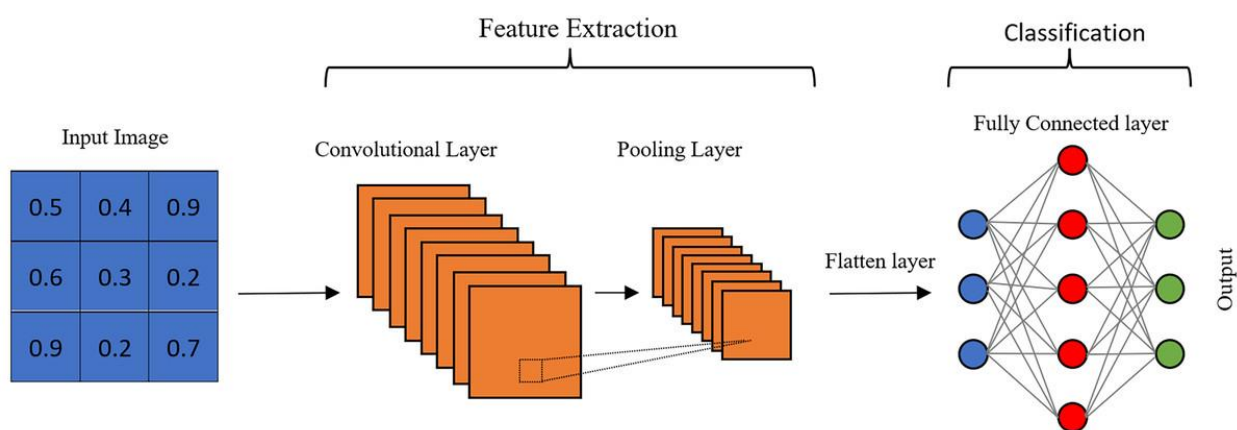


Рисунок 3.16 – Архітектура CCN

Основні компоненти згорткової нейронної мережі:

- Згорткові шари (Convolutional Layers):
 - Виконують операцію згортки, яка виділяє ознаки (особливості) з вхідних даних.
 - У згортці використовується ядро (фільтр), що "ковзає" по вхідному зображенню і обчислює локальні ознаки, такі як контури, текстури або форми.
- Шари активації (Activation Layers):

- Після згортки використовується функція активації (зазвичай ReLU), яка додає нелінійність до моделі, дозволяючи мережі виявляти складні взаємозв'язки в даних.
- Шари підвибірки (Pooling Layers):
 - Використовуються для зменшення розмірності просторових даних, зберігаючи основні ознаки.
 - Найпоширеніший підхід - max pooling, який вибирає максимальне значення в певному регіоні.
- Повністю зв'язані шари (Fully Connected Layers):
 - У фінальній частині мережі використовується кілька щільно пов'язаних шарів для прийняття рішень на основі виділених ознак.
- Dropout і нормалізація (Optional Layers):
 - Dropout використовується для запобігання перенавчанню.
 - Нормалізація шарів (Batch Normalization) стабілізує навчання і пришвидшує збіжність.

Для навчання будуть використані пари зображень: вхідне зображення мапи тропосферного індексу та відповідний цій мапі полігон покриття, який відображає площу покриття станції АІС. Після навчання очікується, що мережа сама зможе прогнозувати полігон покриття за вхідною тропосферною мапою (рис. 3.17).

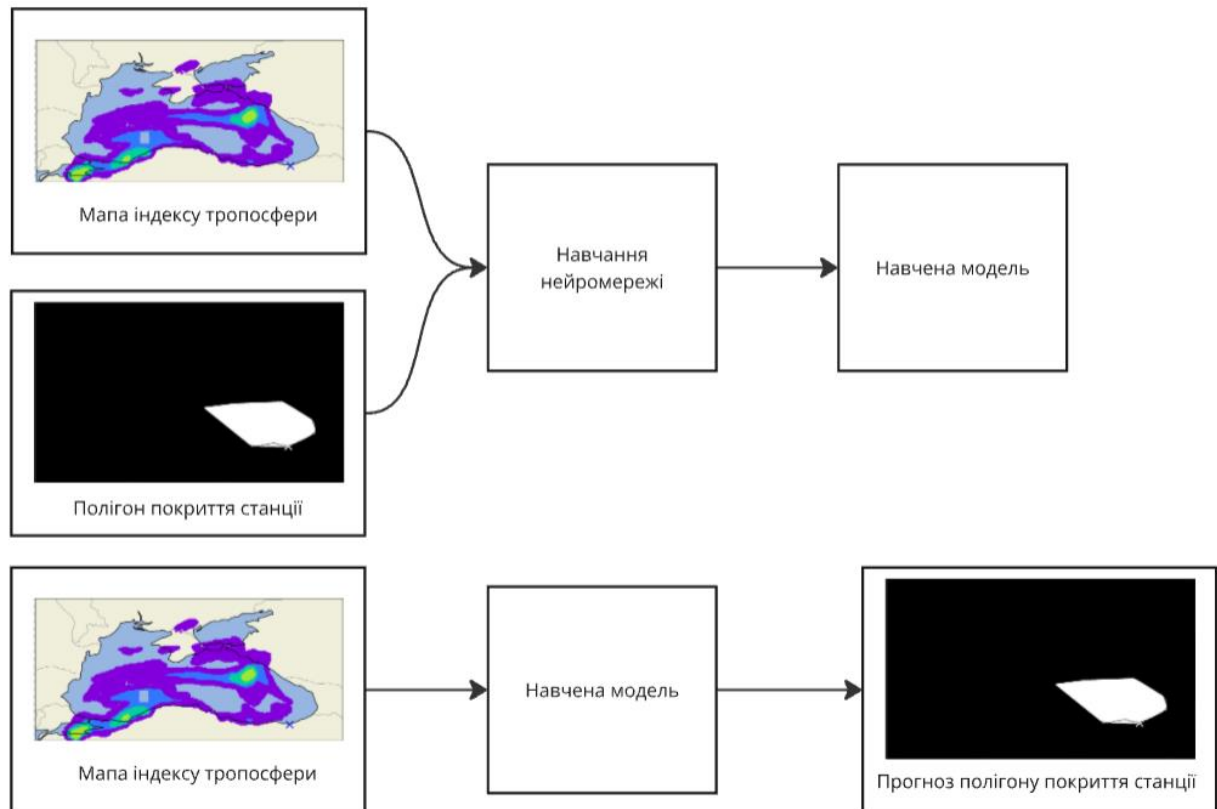


Рисунок 3.17 – Схема навчання мережі

Для цієї задачі підійде сегментація, тобто процес поділу зображення на кілька значущих частин або сегментів для спрощення аналізу або виділення певних об'єктів. У контексті комп'ютерного зору сегментація використовується для точного розпізнавання та класифікації пікселів у зображенні відповідно до їх приналежності до конкретних об'єктів або областей. Вибір пав на DeepLabv3+ згідно зі статтею [20].

DeepLabv3+ є архітектурою нейронної мережі (рис. 3.18), спеціально розробленою для сегментації зображень на семантичні класи. Вона об'єднує найкращі аспекти двох ключових підходів: просторового пірамідального пулінгу (SPP) та структури кодер-декодер.

Вхідне зображення проходить через низку згорткових блоків, поступово зменшуючи розмір зображення та збільшуючи кількість каналів. На цьому етапі вилучаються високорівневі ознаки. Отримані ознаки передаються в модуль SPP, де вони обробляються з використанням атрозної

сепарабельної згортки в різних масштабах. Це дозволяє захопити контекстну інформацію про об'єкти різного розміру. Ознаки, оброблені модулем SPP, подаються на вхід декодера. Декодер поступово збільшує розмір зображення та відновлює деталі, використовуючи також атрозну сепарабельну згортку. На виході декодера отримується карта ймовірностей, де кожному пікселю присвоюється ймовірність належності до певного семантичного класу.

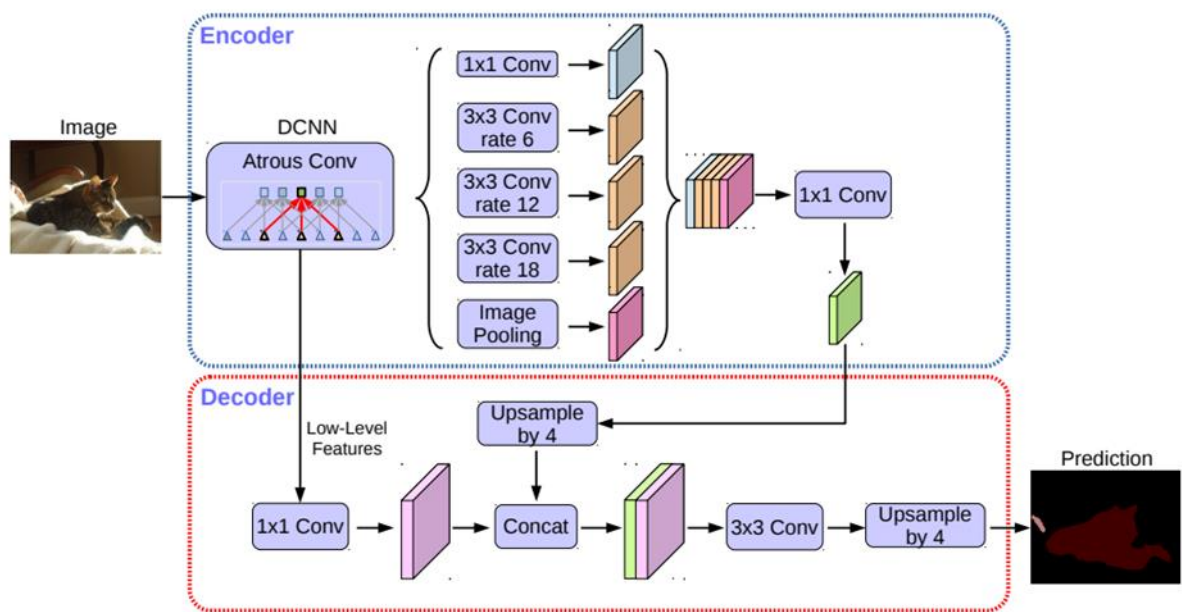


Рисунок 3.18 – Запропонований у статті [20] DeepLabv3+ розширює DeepLabv3 за рахунок використання структури енкодуювальника.

Ключові особливості DeepLabv3+:

- Атрозна сепарабельна згортка: цей тип згортки дозволяє мережі ефективно захоплювати контекстну інформацію в різних масштабах, не збільшуючи обчислювальні витрати. Вона використовується як у модулі SPP, так і в декодері.
- Модуль SPP: завдяки цьому модулю мережа здатна аналізувати вхідне зображення в різних масштабах, що особливо корисно для об'єктів різного розміру.

- Структура кодер-декодер: кодер відповідає за вилучення високорівневих ознак зображення, тоді як декодер відновлює деталі та точні межі об'єктів.
- Висока точність: DeepLabv3+ демонструє передові результати на таких популярних датасетах, як PASCAL VOC 2012 та Cityscapes.

Модель DeepLabv3+ буде використовувати попередньо навчену модель DenseNet121 як базову модель для витягання ознак. В статті [20] приведено детальний опис цієї моделі.

Densely Connected Convolutional Networks (DenseNet) [21] – це тип конволюційних нейронних мереж, який пропонує новий підхід до з'єднання шарів. На відміну від традиційних мереж, де кожен шар отримує вхід лише від попереднього шару, у DenseNet кожен шар отримує вхід від усіх попередніх шарів. Це створює прямі з'єднання між шарами, що призводить до кількох переваг:

- Покращена передача інформації: кожен шар має доступ до градієнтів та функцій активації всіх попередніх шарів, що полегшує поширення інформації та градієнтів через мережу.
- Зменшення проблеми зникнення градієнтів: прямі з'єднання допомагають вирішити проблему зникнення градієнтів, яка часто виникає у глибоких мережах.
- Покращення повторного використання ознак: кожен шар може використовувати ознаки, вивчені попередніми шарами, що дозволяє мережі ефективніше вивчати складні залежності.
- Зменшення кількості параметрів: оскільки кожен шар має менше фільтрів, загальна кількість параметрів мережі зменшується.

Код тренування моделі приведено в додатку Е. Спочатку виконується імпорт бібліотек: `os` - для роботи з файловою системою. `numpy` - для роботи з масивами. `tensorflow` і `keras` - для побудови та навчання нейронної мережі.

					171.6321M.KP.ПЗ.РЗ	Арк.
						72
Змн.	Арк.	№ документа	Підпис	Дата		

sklearn.model_selection - для розділення даних на тренувальний і тестовий набори.

Далі зменшуємо розміри вхідних зображень. Обробка великих зображень (наприклад, 1800x1050) потребує значних обчислювальних ресурсів, особливо якщо ви тренуєте модель на GPU чи CPU з обмеженою пам'яттю. Зменшення розміру до 256x256 значно скорочує кількість пікселів, які модель обробляє. В нашому випадку це понад 28 разів менше даних, що пришвидшує тренування і зменшує споживання пам'яті. Більшість моделей глибокого навчання, зокрема архітектури на основі попередньо навчених мереж (як-от DenseNet121 у нашому випадку), очікують фіксований розмір вхідних даних. Зменшення розміру до 256x256 забезпечує відповідність розміру вхідних зображень розміру, необхідному для коректної роботи базової моделі.

```
def load_images(image_dir, mask_dir, img_height, img_width):
    images = []
    masks = []
    image_files = sorted(os.listdir(image_dir))
    mask_files = sorted(os.listdir(mask_dir))
    for img_file, mask_file in zip(image_files, mask_files):
        img_path = os.path.join(image_dir, img_file)
        mask_path = os.path.join(mask_dir, mask_file)
        img = load_img(img_path, target_size=(img_height, img_width))
        mask = load_img(mask_path, target_size=(img_height, img_width), color_mode="grayscale")
        img = img_to_array(img) /
        mask = img_to_array(mask) /
        images.append(img)
        masks.append(mask)
    return np.array(images), np.array(masks)
images, masks = load_images(input_dir, output_dir, IMG_HEIGHT, IMG_WIDTH)
X_train, X_test, y_train, y_test = train_test_split(images, masks, test_size=0.2, random_state=42)
```

Рисунок 3.19 – Функція для завантаження та нормалізації вхідних зображень та масок

На рисунку 3.19 приведено функцію, в якій images буде містити всі завантажені зображення. Masks буде містити відповідні маски. os.listdir отримує список файлів у директорії а sorted гарантує, що зображення та маски будуть завантажуватися в однаковому порядку. В циклі відбувається

					171.6321М.КР.ПЗ.РЗ	Арк.
						73
Змн.	Арк.	№ документа	Підпис	Дата		

обробка зображень і масок, zip об'єднує два списки (файли зображень і масок) для паралельної ітерації а os.path.join створює повний шлях до файлу. load_img із tensorflow.keras.utils завантажує зображення і змінює його розмір до $\text{img_height} \times \text{img_width}$. Після чого mask = load_img завантажує маску у градаціях сірого (color_mode="grayscale"). img_to_array перетворює зображення з формату PIL у NumPy-масив. Значення кожного пікселя нормалізуються в діапазон [0, 1] шляхом ділення на 255. Дані додаються до списків ,які також перетворюються в NumPy-масиви.

```
def create_deeplabv3plus_model(input_shape):
    base_model = tf.keras.applications.DenseNet121(input_shape=input_shape, include_top=False)
    base_model.trainable = False
    inputs = keras.Input(shape=input_shape)
    x = base_model(inputs, training=False)
    x = layers.Conv2D(256, 3, padding="same", activation="relu")(x)
    x = layers.Conv2D(128, 3, padding="same", activation="relu")(x)
    x = layers.Conv2D(64, 3, padding="same", activation="relu")(x)
    x = layers.UpSampling2D(size=(4, 4), interpolation='bilinear')(x)
    x = layers.Conv2D(32, 3, padding="same", activation="relu")(x)
    x = layers.UpSampling2D(size=(8, 8), interpolation='bilinear')(x)
    outputs = layers.Conv2D(1, 1, activation="sigmoid")(x)
    model = keras.Model(inputs, outputs)
    return model

model = create_deeplabv3plus_model((IMG_HEIGHT, IMG_WIDTH, 3))
model.compile(optimizer="adam", loss="binary_crossentropy", metrics=["accuracy"])
history = model.fit(X_train, y_train, epochs=60, batch_size=8, validation_data=(X_test, y_test))
```

Рисунок 3.20 – Функція для створення DeepLabv3+

Функція (Рис 3.20) створює модель для сегментації, яка приймає RGB-зображення (input_shape = (IMG_HEIGHT, IMG_WIDTH, 3)) і повертає карту ймовірностей сегментів (маску). Функція використовує як базову модель, попередньо натреновану модель DenseNet121 без верхніх шарів (include_top=False), видаляє класифікаційні шари.

inputs = keras.Input(shape=input_shape) Створюється вхідний шар для моделі, який приймає зображення заданої розмірності. Після застосовуються згорткові шари (Conv2D) із фільтрами 256, 128 та 64 для обробки даних після базової моделі. Використовується активаційна функція relu для нелінійності.

					171.6321М.КР.ПЗ.РЗ	Арк.
						74
Змн.	Арк.	№ документа	Підпис	Дата		

padding="same" зберігає розмірність карти ознак після згортки. Додається ще один згортковий шар із 32 фільтрами та збільшення розмірності у 8 разів, щоб отримати остаточну карту ознак із розмірністю, близькою до вхідного зображення.

У вихідному шарі Conv2D із ядром розміром 1x1 з активацією sigmoid генерує карту ймовірностей для кожного пікселя: Значення в діапазоні [0, 1] вказують на ймовірність, що піксель належить до цільового класу (наприклад, білий на масці). Після чого всі шари об'єднуються. Модель компілюється використовуючи оптимізатор Adam.

Алгоритм Adam (Adaptive Moment Estimation) - це популярний оптимізатор для навчання нейронних мереж, який поєднує переваги двох інших методів оптимізації: AdaGrad (адаптивний градієнтний спуск) і RMSProp.

Його особливості:

- Адаптивне навчання для кожного параметра: Adam автоматично налаштовує швидкість навчання для кожного параметра, що забезпечує швидше зближення.
- Моментум (momentum): Adam зберігає інформацію про напрямок градієнта, завдяки чому рух у відповідному напрямку стабільніший.

Після чого модель тренується: `history = model.fit(X_train, y_train, epochs=60, batch_size=8, validation_data=(X_test, y_test))` - X_train, y_train - навчальний набір зображень і відповідних масок. X_test, y_test - тестовий набір для валідації. epochs=40: Кількість епох навчання. batch_size=8: розмір міні-пакета. validation_data: Валідаційний набір використовується для оцінки моделі після кожної епохи.

На рисунку 3.21 приведено приклад того як модель прогнозує покриття після навчання.

					171.6321M.KP.ПЗ.РЗ	Арк.
						75
Змн.	Арк.	№ документу	Підпис	Дата		

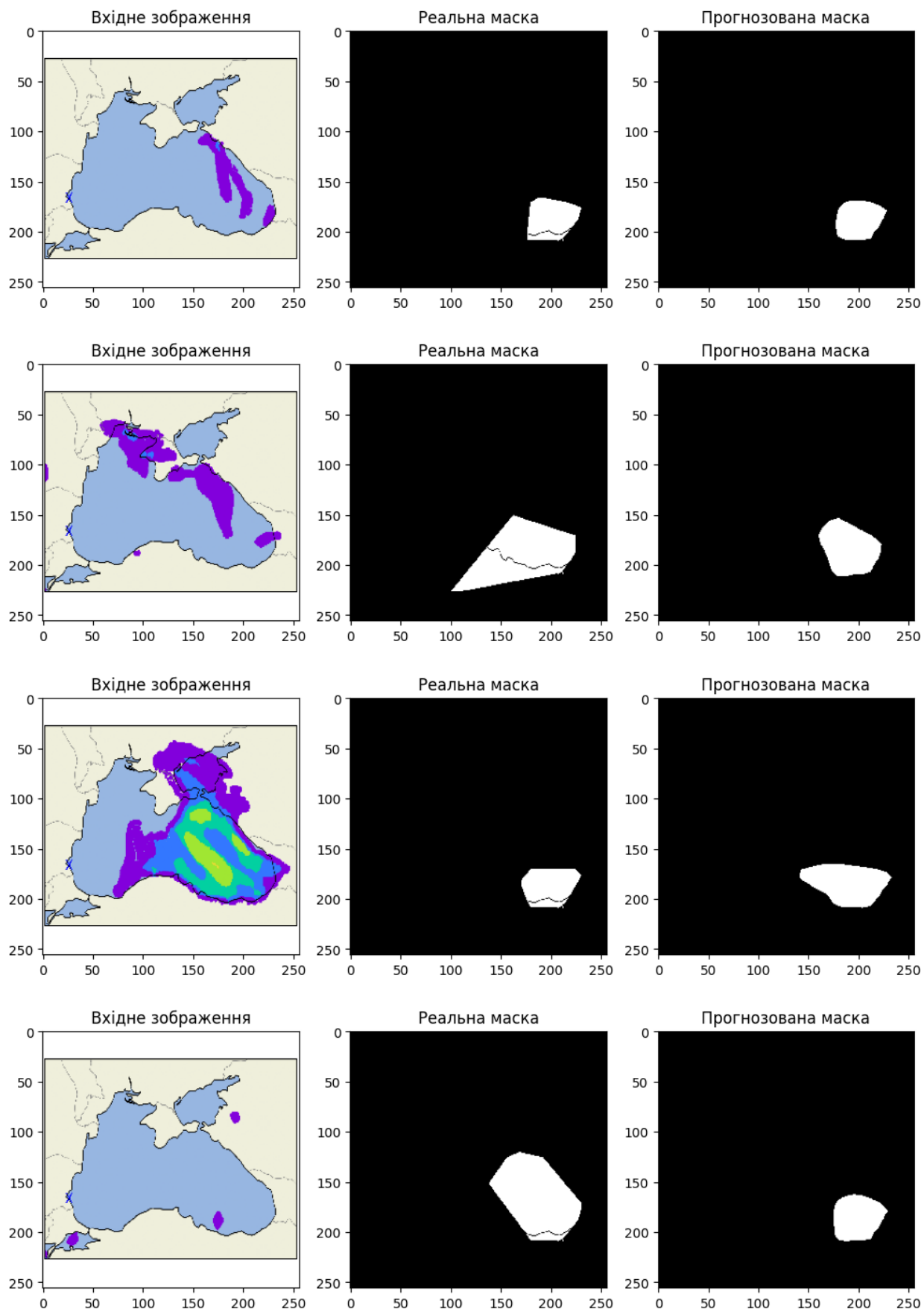


Рисунок 3.21 – Приклад прогнозу покриття

Спробуємо оцінити модель на нових даних, які не використовувались при навчанні нейромережі для цього напишемо скрипт, який приведено у додатку Є. Цей код аналізує ступінь збігу між двома наборами бінарних зображень (масок) у двох папках. Зокрема, він обчислює подібність між відповідними файлами з двох папок, щоб оцінити, наскільки маски покриття однієї системи (прогноз нейронної мережі) збігаються з іншими (еталонними даними).

`calculate_polygon_similarity` - функція, яка: завантажує два зображення за вказаними шляхами у відтінках сірого; конвертує ці зображення у бінарні маски (чорне -> 0, біле -> 255); перевіряє, чи зображення мають однаковий розмір. Якщо розміри різні, викликається помилка. Обчислює коефіцієнт подібності між пікселями двох зображень. Перетворює коефіцієнт подібності в проценти та повертає його.

`process_images` - функція, яка: отримує списки файлів у двох папках; визначає спільні файли (з однаковими назвами) у двох папках. Для кожної пари спільних файлів викликає `calculate_polygon_correlation` для обчислення кореляції. Зберігає значення кореляції та імена файлів. Обчислює середній ступінь збігу між усіма парами масок. Результат приведено в таблиці 3.1
Середня ступінь збігу для всіх прогнозів: 71.03%

Таблиця 3.1 Результат оцінки

Назва	Ступінь збігу
00_00_eeu_28_03_2023.png	56.59%
21_00_eeu_05_04_2023.png	51.85%
06_00_eeu_06_04_2023.png	59.79%
21_00_eeu_22_03_2023.png	88.81%
03_00_eeu_23_03_2023.png	81.91%
21_00_eeu_15_04_2023.png	67.10%
18_00_eeu_01_04_2023.png	91.19%

За цими даними можна побудувати діаграму (рис 3.22).

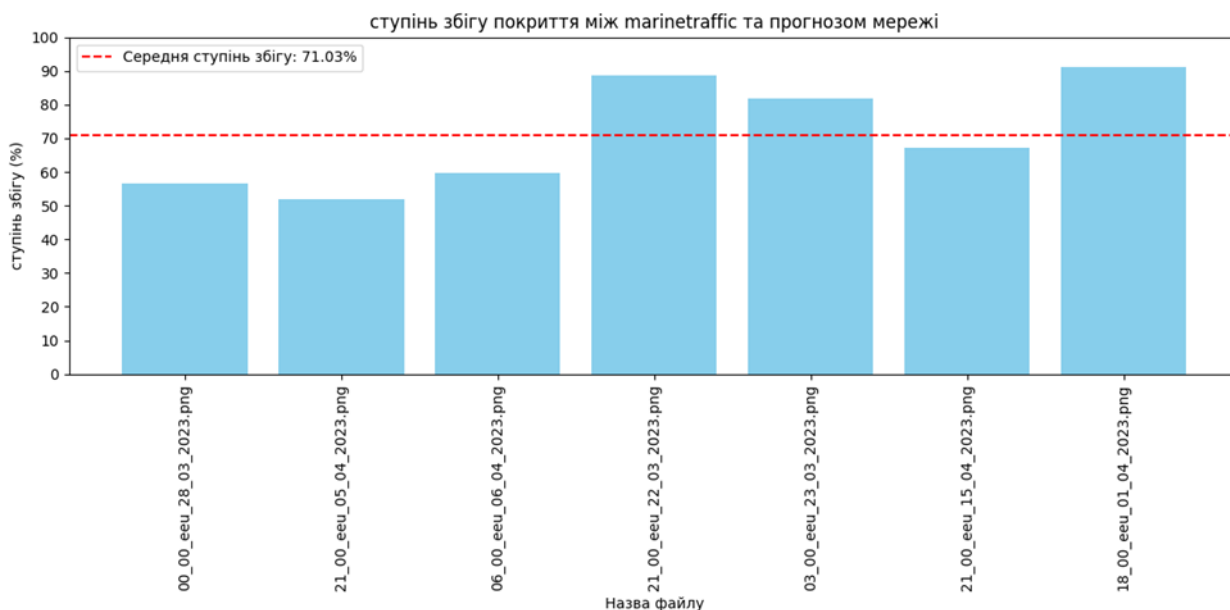


Рисунок 3.22 – Діаграма ступеню збігу маски покриття між даними з marinetraffic та прогнозом мережі

3.4. Залежність кількості прийнятих АІС-повідомлень від площі зони покриття

Одним із важливих параметрів, що характеризують роботу АІС-станцій, є кількість прийнятих АІС-повідомлень. Тож однією з цілей є встановлення математичної залежності між площею зони покриття та кількістю прийнятих повідомлень. Це дозволить оцінити ефективність роботи станції та оптимізувати її функціонування.

Для досягнення поставленої мети можна використати метод лінійної регресії, який дозволяє встановити лінійну залежність між зазначеними параметрами. Вхідні дані було отримано з marine traffic (рис 3.23), вони містять інформацію про площу зон покриття та кількість прийнятих сигналів.

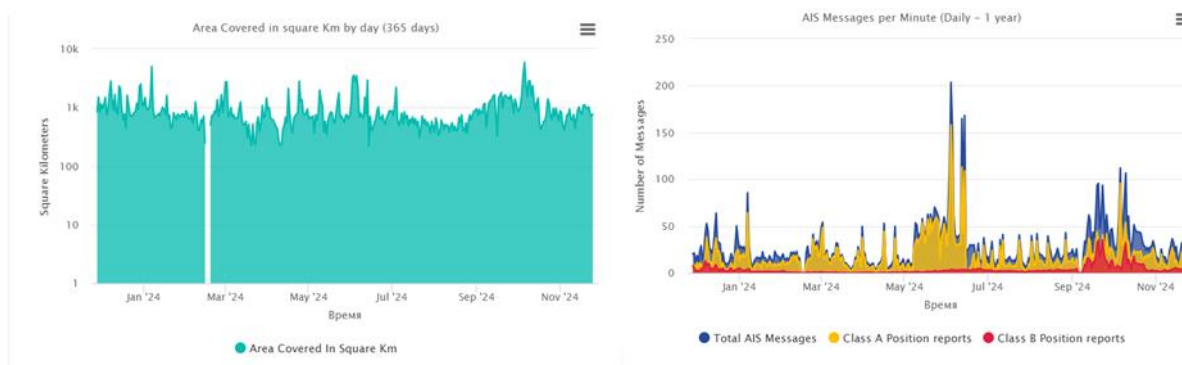


Рисунок 3.23 – Площа покриття та кількість прийнятих сигналів за рік

Код побудови моделі приведено в додатку Ж. В цьому коді клас LinearRegression використовується для створення моделі лінійної регресії, яка дозволяє прогнозувати залежність між незалежною змінною (або змінними) та залежною змінною шляхом побудови лінії, що найкраще підходить під дані. Лінійна регресія є базовою технікою регресійного аналізу.

Спочатку код підготовлює дані формуючи три списки: дат, площ та кількостей сигналів. Після чого дані конвертуються в масиви NumPy для подальшої обробки. Змінна X містить масив площ (незалежна змінна), а y - масив кількостей сигналів (залежна змінна). Далі створюється модель лінійної регресії, яка навчається на даних, мета навчання - знайти оптимальні значення коефіцієнтів лінійної функції, які мінімізують середньоквадратичну похибку.

Після чого розраховуються метрики якості моделі: Середньоквадратична похибка (MSE): показує середнє квадратичне відхилення прогнозів від реальних значень. Чим менше значення MSE, тим краще модель. Коефіцієнт детермінації (R^2): показує, яка частка дисперсії залежної змінної пояснюється моделлю. Чим ближче R^2 до 1, тим краще модель.

Останнім кроком є побудова графіку залежності кількості сигналів від площі. На графіку (рис. 3.24) зображені реальні дані та лінія регресії.

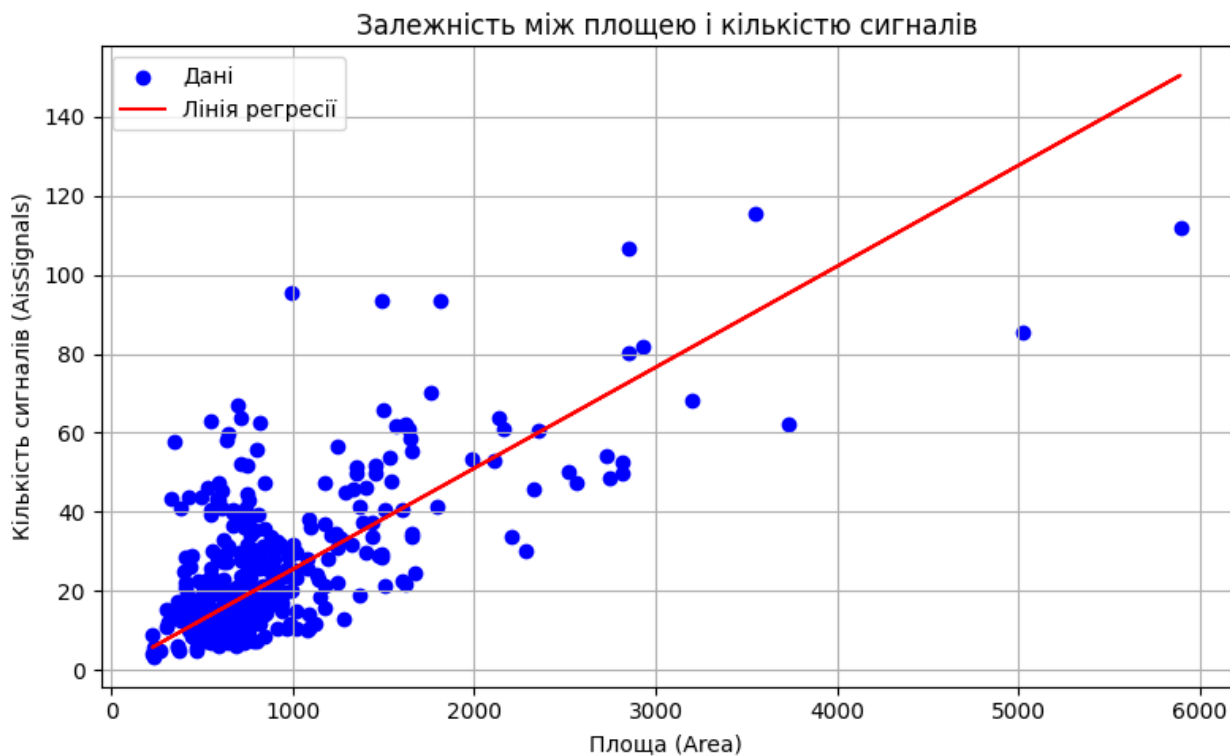


Рисунок 3.24 – Графік залежності кількості сигналів від площі

Коефіцієнт детермінації (R^2) у даної моделі 0.45.

Тепер використовуючи цю модель можна спрогнозувати кількість прийнятих сигналів для прогнозованих нейромережею площ покриття(рис. 3.25)

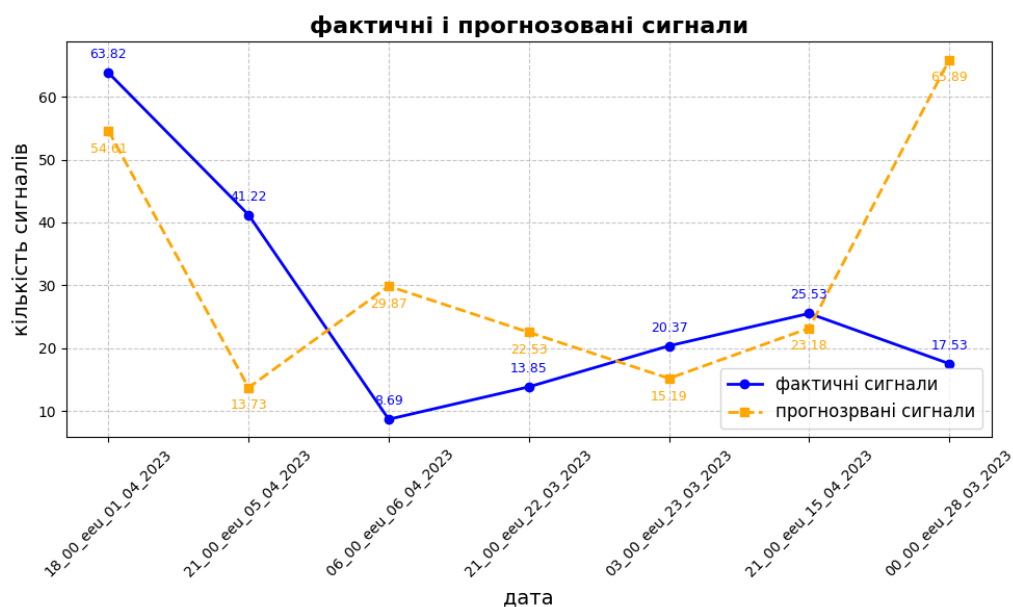


Рисунок 3.25 Графік реальної кількості сигналів та прогнозованої

Висновок

У розділі представлено приклад розробки нейромережевої моделі для прогнозування області покриття АІС-станції на основі карт індексу тропосфери Хелберна (Herburn Tropo Index, HTI).

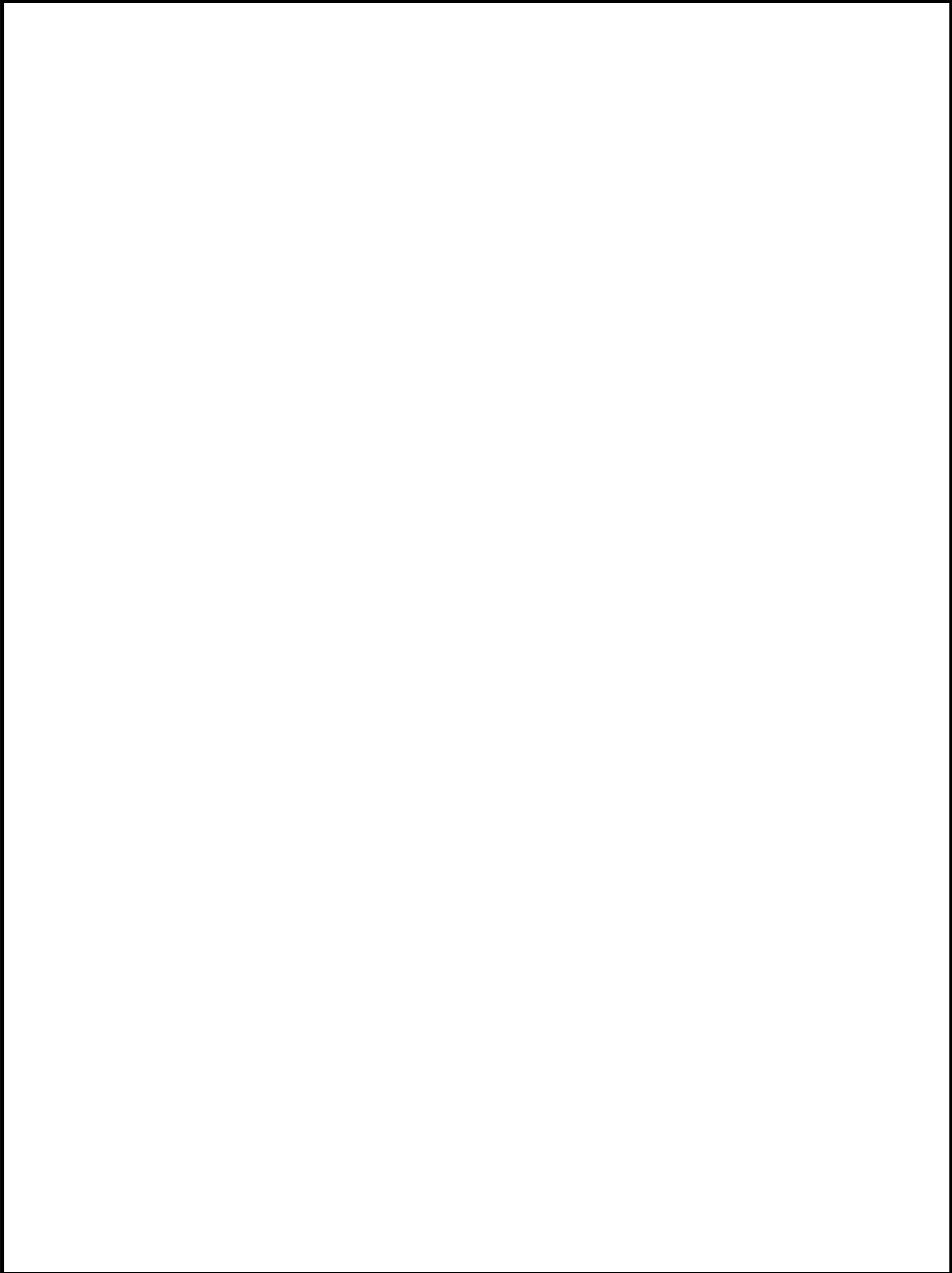
Представлено збір і підготовку даних, а також формування наборів для навчання, тестування та валідації моделі.

Розглянуто архітектуру нейронних мереж, зокрема згорткові нейронні мережі (CNN). Розроблено структуру нейронної мережі, адаптовану до задачі прогнозування покриття. увагу приділено оптимізації гіперпараметрів, функції активації та методи регуляризації.

Проведено навчання моделі на основі підготовлених даних. Для тестування використано відкладену частину вибірки, що дозволило оцінити точність і надійність прогнозів. У процесі було визначено, як ефективно модель враховує варіації тропосферних умов.

Також у розділі проаналізовано залежність кількості прийнятих АІС-повідомлень від площі зони покриття. Побудовано модель лінійної регресії, яка демонструє цю залежність, Показано приклад використання цієї моделі для прогнозування кількості сигналів на основі прогнозів площі покриття, які отримано від нейромережі.

					171.6321М.КР.ПЗ.РЗ	Арк.
						81
Змн.	Арк.	№ документу	Підпис	Дата		



					171.6321М.КР.ПЗ.Р4			
Змн.	Арк.	№ документу	Підпис	Дата				
Розробник		Вишневський В. В.			Охорона праці	Літ.	Арк.	Аркуші
Консультант		Губальцев А. М.					82	
Н. контр.		Добрінова Л. П.				НУК імені адмірала Макарова		
Керівник		Дьяконов О. С.						
Зав. каф.		Рябенський В. М.						

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

4.1. Аналіз шкідливих та небезпечних факторів на робочих місцях

Згідно з наказом [22] робота з комп'ютерною технікою може створювати різноманітні ризики для здоров'я працівників. Вплив шкідливих факторів зумовлений тривалістю перебування за комп'ютером, рівнем організації робочого місця, а також індивідуальними особливостями користувача. Детальніше розглянемо основні загрози:

Напруження органів зору: Тривала робота за монітором є основною причиною зорового перенапруження, яке відоме як комп'ютерний зоровий синдром (КЗС). До його основних проявів належать:

- Відчуття сухості, печіння та почервоніння очей.
- Зниження гостроти зору.
- Головний біль, спричинений перевтомою очей.

Причини цих симптомів полягають у недостатньому освітленні робочого місця, невідповідних параметрах монітора (наприклад, низька роздільна здатність, мерехтіння екрана) та неправильній відстані від очей до екрана.

Статичне навантаження та неправильна постава: Тривале сидіння за комп'ютером часто супроводжується:

- Неправильною поставою (сутулістю, нахилом голови вперед, закругленням плечей).
- Надмірним статичним напруженням м'язів шиї, спини, плечей і рук.

Ці чинники можуть призводити до розвитку таких хронічних захворювань:

- Остеохондроз шийного та грудного відділів хребта.
- Синдром зап'ястного каналу, викликаний тривалим використанням клавіатури та миші.
- Порушення кровообігу в нижніх кінцівках через здавлювання судин у статичній позі.

					171.6321М.КР.ПЗ.Р4	Арк.
						83
Змн.	Арк.	№ документа	Підпис	Дата		

Психоемоційне навантаження: Робота з комп'ютером часто потребує високої концентрації уваги, швидкого прийняття рішень та багатозадачності. Це може спричиняти:

- Хронічний стрес через надмірну інтенсивність роботи.
- Емоційне виснаження внаслідок монотонності завдань.
- Складнощі із засинанням через порушення біологічних ритмів, пов'язані з використанням екранів у вечірній час.

Вплив електромагнітного випромінювання: Хоча сучасні монітори (наприклад, LED або LCD) генерують мінімальне електромагнітне випромінювання, воно все ж може мати негативний вплив за умов тривалого контакту. Крім того, працюючі комп'ютери нагрівають повітря, що сприяє утворенню мікропили і сухості в приміщенні.

Невідповідний мікроклімат робочого місця: Температура, вологість і якість повітря на робочому місці значно впливають на комфорт працівника. Основними негативними аспектами можуть бути:

- Підвищена сухість повітря через тривалу роботу кондиціонерів чи нагрівачів.
- Недостатня вентиляція, яка призводить до накопичення вуглекислого газу та погіршення самопочуття.
- Температурні перепади, що створюють додаткове навантаження на організм.

Шумовий вплив: Навіть низький рівень шуму, створений вентилятором комп'ютера, може викликати дискомфорт. Тривала робота в умовах постійного звуку сприяє втомі, зниженню концентрації уваги і може призводити до головного болю.

4.2. Вимоги до організації робочого місця

Правильна організація робочого місця є ключовим елементом для забезпечення комфортної, безпечної та ефективної роботи з комп'ютером. Вона сприяє зниженню негативного впливу на здоров'я та підвищенню

					171.6321M.KP.ПЗ.Р4	Арк.
Змн.	Арк.	№ документа	Підпис	Дата		84

продуктивності праці. Розглянемо основні аспекти організації робочого місця:

Ергономіка робочого місця:

- Робочий стіл повинен бути достатньо широким, щоб розмістити монітор, клавіатуру, мишу та інші необхідні предмети. Рекомендована висота столу – 70–75 см. Матеріал стільниці має бути матовим, щоб уникнути відблисків.
- Стілець повинен бути з регульованою висотою, підлокітниками та підтримкою для попереку. Висота стільця налаштовується так, щоб ноги працівника стояли на підлозі або підставці під кутом 90 градусів у колінному суглобі.

Розташування обладнання

- Відстань між очима працівника та екраном має бути 50–70 см.
- Верхній край екрана має бути трохи нижче рівня очей (на 10–15 градусів).
- Монітор слід розташовувати прямо перед користувачем, щоб уникнути поворотів голови та ший.
- Клавіатура повинна бути розташована на рівні ліктів, щоб зап'ястя залишалися прямими під час роботи.
- Миша має бути поруч із клавіатурою, щоб уникнути перенапруження рук.
- Для запобігання застою крові в ногах слід використовувати підставку для ніг, особливо якщо ноги не дістають до підлоги.

Освітлення робочого місця: Правильне освітлення зменшує напруження очей і сприяє комфортній роботі. Основні вимоги:

- Природне освітлення: Джерело природного світла (вікно) повинно розташовуватися збоку від працівника – переважно зліва, якщо працівник правша, і справа – для шульги. Це зменшує утворення відблисків на екрані.

					171.6321М.КР.ПЗ.Р4	Арк.
						85
Змн.	Арк.	№ документу	Підпис	Дата		

- Штучне освітлення: Освітлення має бути рівномірним, з використанням настільної лампи з матовим плафоном, що направляє світло на робочу поверхню, але не на екран.

Необхідно уникати прямого потрапляння світла на монітор. За потреби використовуються штори, жалюзі або спеціальні антивідблискові екрани.

Вимоги до мікроклімату: Комфортний мікроклімат робочого місця забезпечує оптимальне самопочуття працівника, що сприяє зниженню втоми та підвищенню продуктивності:

- Оптимальна температура для роботи за комп'ютером становить 18–22°C.
- Рекомендована відносна вологість – 40–60%. Для підтримання рівня вологості можна використовувати зволожувачі повітря.
- Робоче приміщення повинно регулярно провітрюватися, щоб забезпечувати приплив свіжого повітря та знижувати концентрацію вуглекислого газу.

Акустичний комфорт: Хоча робота за комп'ютером зазвичай супроводжується мінімальним шумом, важливо враховувати:

- Шум від техніки (вентилятори, системні блоки, принтери) не повинен перевищувати 40–50 дБ. Якщо шум перевищує ці значення, слід передбачити шумоізоляцію приміщення або використання малошумного обладнання.

Підтримання чистоти на робочому місці: Комп'ютерне обладнання притягує пил, що може впливати на якість повітря та функціональність техніки.

- Регулярне прибирання пилу з поверхонь монітора, клавіатури та інших елементів.
- Використання антистатичних серветок для очищення екрана.

					171.6321М.КР.ПЗ.Р4	Арк.
						86
Змн.	Арк.	№ документа	Підпис	Дата		

4.3. Заходи для запобігання шкідливим факторам

Для зменшення підвищеного рівня статичного струму під час роботи з ПК рекомендується дотримуватись таких заходів:

- Заземлення: Переконайтеся, що всі компоненти комп'ютера, включаючи корпус, монітор та периферійні пристрої, заземлені належним чином. Використовуйте заземлені розетки або антистатичні мати для захисту обладнання.
- Антистатичні пристрої: Під час складання або ремонту комп'ютера використовуйте антистатичні мати чи браслети, щоб уникнути накопичення статичного заряду.
- Уникнення накопичення зарядів: Намагайтеся мінімізувати рухи або тертя об пластикові поверхні (наприклад, килимки), які можуть сприяти генерації статичного заряду.
- Вологе прибирання: Проводьте регулярне очищення робочого місця вологою тканиною або спеціальними антистатичними засобами, щоб знизити рівень статичного заряду.
- Електростатичні розрядники: Розгляньте встановлення пристроїв, які автоматично розряджають статичний заряд у приміщенні.

Заходи для зниження фізичних навантажень при роботі з ПК:

- Ергономічне обладнання: Використовуйте спеціальні ергономічні меблі, наприклад, стіл і стілець із підтримкою спини, а також регульовані клавіатуру й мишу, щоб адаптувати їх до вашої пози.
- Правильна постава: Дотримуйтесь правильної постави, тримаючи спину прямою, плечі розслабленими, а стегна та коліна – на одному рівні.
- Регулярні паузи: Робіть перерви під час роботи, виконуйте легкі вправи для розтягування м'язів ший, плечей і кистей.
- Вправи для очей: Регулярно проводьте вправи для зняття напруги очей, наприклад, фокусуйте погляд на віддалених об'єктах.

					171.6321M.KP.ПЗ.Р4	Арк.
						87
Змн.	Арк.	№ документа	Підпис	Дата		

- Фізична активність: Включайте щоденні вправи, прогулянки чи розтяжки, щоб зменшити м'язову напругу та покращити кровообіг.

Для запобігання перевтомі зорового апарату:

- Оптимальна позиція та освітлення: Правильно налаштуйте висоту й нахил монітора, забезпечте рівномірне та достатнє освітлення робочого місця.
- Налаштування екрана: Відрегулюйте яскравість, контрастність та кольорову температуру екрана для комфортної роботи.
- Фільтри синього світла: Використовуйте фільтри синього світла, щоб знизити вплив шкідливого випромінювання.
- Режим роботи: Забезпечте регулярний відпочинок для очей та якісний сон, що сприятиме їхньому здоров'ю.

Для усунення проблем зі зниженою іонізацією повітря:

- Іонізатори повітря: Використовуйте пристрої для підтримання балансу зарядів у повітрі, що зменшить статичний заряд.
- Очищення поверхонь: Регулярно протирайте обладнання вологою тканиною або спеціальними засобами для зменшення накопичення заряду.
- Антистатичні матеріали: Використовуйте антистатичні килимки, підставки або покриття для нейтралізації статичного заряду.
- Оптимальний рівень вологості: Застосовуйте зволожувачі повітря, щоб підтримувати комфортну вологість у приміщенні.

Для зменшення впливу електромагнітного випромінювання:

- Розташування робочого місця: Тримайте ПК на безпечній відстані від місць тривалого перебування та дотримуйтесь правил розміщення монітора.
- Екранування: Використовуйте екрани чи фільтри для зменшення відблисків і випромінювання.

					171.6321M.KP.ПЗ.Р4	Арк.
						88
Змн.	Арк.	№ документа	Підпис	Дата		

- Заземлення: Перевірте, чи обладнання заземлене належним чином, щоб уникнути електростатичних розрядів.
- Екрановані кабелі: Використовуйте кабелі з екрануванням для зменшення інтерференцій.
- Бездротові пристрої: По можливості використовуйте бездротові технології, щоб зменшити рівень випромінювання від пристроїв.

4.4. Домедична допомога постраждалим від електричного струму

Робота за комп'ютером може бути небезпечною через ризик ураження електричним струмом. Це може статися через несправності електропроводки, неправильне використання обладнання або недостатню ізоляцію. Електротравми бувають різного ступеня тяжкості — від незначних до критичних, і можуть серйозно вплинути на здоров'я.

Згідно з Порядком надання домедичної допомоги постраждалим від електричного струму та блискавки (zareestrovanim u Ministerstvi yustitsii Ukraini 7 lipnya 2014 roku, № 775/25552)[1], електротравма — це ушкодження, спричинене впливом потужного електричного струму або розрядом блискавки. Документ визначає необхідні дії для надання першої допомоги в таких ситуаціях.

Вплив струму залежить від його сили, тривалості, шляху проходження через тіло та інших обставин. Електротравми можуть викликати:

- Місцеві пошкодження: наприклад, опіки в точках контакту.
- Загальні порушення організму: зупинка серця чи дихання, дисфункція внутрішніх органів.

Для запобігання електротравмам важливо дотримуватися правил безпеки:

- Використовувати обладнання з належним заземленням.
- Регулярно перевіряти електропроводку.
- Застосовувати сертифіковану техніку.

					171.6321M.KP.P3.P4	Арк.
						89
Змн.	Арк.	№ документа	Підпис	Дата		

Дії під час допомоги мають бути швидкими, але продуманими, щоб захистити як постраждалого, так і себе.

- Перевірка безпеки Переконайтеся, що джерело струму більше не становить загрози. У разі небезпеки ізолюйте себе від джерела (наприклад, оголених дротів).
- Припинення дії струму
 - Вимкніть електроживлення за допомогою рубильника чи вимикача.
 - Якщо це неможливо, використовуйте ізолюючі предмети (суху палицю, пластикову трубу), щоб відсунути джерело струму від потерпілого.
- Огляд постраждалого
 - Перевірте свідомість: спробуйте звернутися або злегка потрусити його за плечі.
 - Оцініть дихання (рух грудної клітки, звуки дихання) і пульс.
- Виклик екстреної допомоги негайно зателефонуйте в екстрену службу та повідомте:
- місце події;

					171.6321М.КР.ПЗ.Р4	Арк.
						90
Змн.	Арк.	№ документу	Підпис	Дата		

ВИСНОВКИ

У роботі виконано глибокий аналіз особливостей роботи автоматичної ідентифікаційної системи (AIC). Розглянуто вплив тропосферних факторів на розповсюдження УКХ-хвиль і формування специфічних хвилеводів, таких як підняті, поверхневі та випаровувальні хвилеводи. Виявлено, що ці явища значно впливають на зону покриття AIC

Особливу увагу приділено використанню тропосферних карт індексу Хепберна (НТІ) для прогнозування площі покриття AIC-станцій. Запропоновано нейромережеву модель, побудовану на основі згорткових нейронних мереж (CNN), що здатна передбачати зони дії AIC, враховуючи змінні тропосферні умови. Модель пройшла навчання, тестування та валідацію, що дозволило підтвердити її здатність прогнозувати покриття та враховувати вплив атмосферних аномалій.

Додатково проаналізовано зв'язок між площею покриття станції та кількістю прийнятих AIC-повідомлень. Побудовано лінійну регресійну модель, яка демонструє цю залежність, і наведено приклади її використання для прогнозування кількості сигналів на основі результатів нейромережевого прогнозу площі покриття.

Таким чином, зроблено вагомий внесок у розуміння впливу тропосферних умов на функціонування AIC, а також запропоновано інструменти для прогнозування зон покриття та завантаженості VDL.

					171.6321М.КР.ПЗ.Висновки	Арк.
						91
Змн.	Арк.	№ документу	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The Nautical Institute. How AIS works and what it does. Nautical Institute [Електронний ресурс] Режим доступу: <https://www.nautinst.org/resources-page/how-ais-works-and-what-it-does.html>
2. Міжнародна морська організація (ІМО) [Електронний ресурс] Режим доступу: <https://www.imo.org/>.
3. ITU-R M.1371-5 (02/2014). Technical characteristics for an automatic identification system using time division multiple access in the VHF maritime mobile frequency band. Geneva: International Telecommunication Union.
4. Топалов, В. П., Торський, В. Г. (2002). Конвенція СОЛАС-74. Основні положення і коментарі (Практичний посібник). Одеса: 288 с.
5. Ground wave propagation [Електронний ресурс] Режим доступу: <https://vu2nsb.com/radio-propagation/ground-wave-propagation/>
6. ITU-R P.834-9 (12/2017) Effects of tropospheric refraction on radiowave propagation.
7. Atmospheric impact on VHF Radio Propagation [Електронний ресурс] Режим доступу: <https://vu2nsb.com/radio-propagation/free-space-propagation/atmospheric-vhf-radio-propagation/>
8. Tropospheric DX Modes [Електронний ресурс] Режим доступу: <https://www.dxinfocentre.com/propagation/tr-modes.htm>
9. Diakonov, O., Ikhsanov, Sh. M., & Khudov, H. (2023). Automatic Identification System VHF Data Link Loading under Anomalous Tropospheric VHF Propagation. International Conference on Engineering, Environmental Science. 19 October 2023.
10. ALA Recommendation A-124 Appendix 18 – VDL Loading Management, December 2011.
11. Discussion on AIS coverage and VDL loading. Baltic Marine Environment Protection Commission. Expert Working Group for Mutual Exchange and Deliveries of AIS & Data, 2016.

					171.6321М.КР.ПЗ.Висновки	Арк.
						92
Змн.	Арк.	№ документа	Підпис	Дата		

12. Recommendation ITU-R M.1371-5 (02/2014) Technical characteristics for an automatic identification system using time division multiple access in the VHF maritime mobile frequency band.

13. Requests: HTTP for Humans [Електронний ресурс] Режим доступу: <https://requests.readthedocs.io/en/latest/>

14. Selenium automates browsers [Електронний ресурс] Режим доступу: <https://www.selenium.dev/>

15. Rasterio: access to geospatial raster data [Електронний ресурс] Режим доступу: <https://rasterio.readthedocs.io/en/stable/>

16. Documentation for EOmaps [Електронний ресурс] Режим доступу: <https://eomaps.readthedocs.io/en/v8.1.1/>

17. OpenCV - Open Computer Vision Library [Електронний ресурс] Режим доступу: <https://opencv.org/>

18. An end-to-end platform for machine learning [Електронний ресурс] Режим доступу: <https://www.tensorflow.org/>

19. Keras: Deep Learning for humans [Електронний ресурс] Режим доступу: <https://keras.io/>

20. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation.

21. Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. Densely connected convolutional networks // IEEE Conference on Computer Vision and Pattern Recognition.1 – 2017. – P. 2261-2269.

22. Міністерство соціальної політики України. (2018). Наказ № 207 від 14.02.2018 р. Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями.

					171.6321М.КР.ПЗ.Висновки	Арк.
						93
Змн.	Арк.	№ документу	Підпис	Дата		

ДОДАТОК А

Коди програм для скрапінгу, та збору даних

Програма для скрапінгу marine traffic

```
import argparse
import time
import pytz
from datetime import datetime
import os
import re
import codecs
import json
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
# See PyCharm help at https://www.jetbrains.com/help/pycharm/

#create parser
parser = argparse.ArgumentParser()
#database name
parser.add_argument('--path',required=True, help='path to work directory')

parser.print_help()
# get arguments
args = parser.parse_args()

#print arguments
print(args)

#get current UTC time
current_time_str = datetime.now(tz=pytz.UTC).strftime("%d_%m_%Y_%H_00")
time_str = datetime.now(tz=pytz.UTC).strftime("%H_00")
sub_dir_name_str = datetime.now(tz=pytz.UTC).strftime("%d_%m_%Y")

#attempts
max_attempts = 5
current_attempt = 0
#set failed flag to enter first iteration
failed_flag = True

options = webdriver.ChromeOptions()
options.add_argument("--disable-blink-features=AutomationControlled")

s = Service()

while (current_attempt<max_attempts and failed_flag == True):
    #clear failed flag
    failed_flag = False
    #increment attempts
    current_attempt=current_attempt+1
    #create webdriver
    driver = webdriver.Chrome(service=s, options=options)

    driver.execute_cdp_cmd('Page.addScriptToEvaluateOnNewDocument', {
        'source': '''
            delete window.cdc_adoQpoasnfa76pfcZLmcfl_Array;
            delete window.cdc_adoQpoasnfa76pfcZLmcfl_Promise;
            delete window.cdc_adoQpoasnfa76pfcZLmcfl_Symbol;'''
    })
```

					171.6321М.КР.ПЗ.Додатки	Арк.
						94
Змн.	Арк.	№ документа	Підпис	Дата		

```

        delete window.cdc_adoQpoasnfa76pfcZLmcfl_JSON;
        delete window.cdc_adoQpoasnfa76pfcZLmcfl_Proxy;
        delete window.cdc_adoQpoasnfa76pfcZLmcfl_Object;
        '''

    })

    #obtain parent window handle
    p= driver.window_handles[0]

main_url="https://www.marinetraffic.com/en/ais/home/centerx:34.3/centery:44.1
/zoom:6"

map_urls=["https://www.marinetraffic.com/getData/get_data_json_4/z:6/X:18/Y:1
1/station:0","https://www.marinetraffic.com/getData/get_data_json_4/z:6/X:18/
Y:12/station:0","https://www.marinetraffic.com/getData/get_data_json_4/z:6/X:
19/Y:11/station:0","https://www.marinetraffic.com/getData/get_data_json_4/z:6
/X:19/Y:12/station:0"]

    #make working directory
    try:
        os.mkdir(args.path)
    except OSError as error:
        print(error)

    #make subdirectory
    try:
        os.mkdir(args.path+"\\ "+sub_dir_name_str)
    except OSError as error:
        print(error)

    #make subdirectory
    try:
        os.mkdir(args.path+"\\ "+sub_dir_name_str+"\\ "+ "map")
    except OSError as error:
        print(error)

    #open main window
    driver.maximize_window()
    driver.get(main_url)
    #wait 10 seconds
    time.sleep(5)

    i=1
    for url in map_urls:
        #make subdirectory for station
        try:
            os.mkdir(args.path + "\\ " + sub_dir_name_str+ "\\ " + "map" + "\\ "
+ time_str)
        except OSError as error:
            print(error)
        print("Scraping data of MT map")
        #open new tab
        driver.execute_script("window.open('');")
        #switch to new tab
        driver.switch_to.window(driver.window_handles[1])
        # open station_map_url
        driver.get(url)
        time.sleep(2)
        n=os.path.join(args.path+"\\ "+sub_dir_name_str+"\\ "+ "map" + "\\ "
+time_str,"map_" +str(i)+ ".json")
        with codecs.open(n, "w", "utf?8") as f:
            h = driver.page_source

```

					171.6321М.КР.ПЗ.Додатки	Арк.
						95
Змн.	Арк.	№ документи	Підпис	Дата		

```

        #remove html tags
        result = re.sub(r'<.*?>', '', h)
        result = (json.dumps(json.loads(result), indent=4))
        f.write(result)
        # check is OK
        if (result.find('Access Denied') != -1):
            failed_flag=True
    #close tab
    time.sleep(1)
    driver.close()
    # switch to parent window
    driver.switch_to.window(p)
    i=i+1

#f.close()

time.sleep(2)
driver.quit()

```

Програма для скрапінгу карт тропосферного індексу

```

import argparse
import requests
import time
import pytz
from datetime import datetime, timedelta
import os

#header for request
headers = {"User-Agent": "Mozilla/5.0 (X11; Linux x86_64; rv:60.0)
Gecko/20100101 Firefox/60.0",
           "Accept":
           "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8",
           "Accept-Language": "en-US,en;q=0.9"
          }

image_url='https://www.dxinfocentre.com/tr_map/fcst/eeu';

#create parser
parser = argparse.ArgumentParser()
#database name
parser.add_argument('--path',required=True, help='path to work directory')

parser.print_help()
# get arguments
args = parser.parse_args()

#print arguments
print(args)

#get current utc time
#add one day
sub_dir_name_str =
(datetime.now(tz=pytz.UTC)+timedelta(days=1)).strftime("%d_%m_%Y")

#create working directory

```

					171.6321М.КР.ПЗ.Додатки	Арк.
						96
Змн.	Арк.	№ документа	Підпис	Дата		

```

file_numbers=list(range(12,39,3))
#add leading zeros
# result_1 = str(num).zfill(3)

#make working directory
try:
    os.mkdir(args.path)
except OSError as error:
    print(error)

#make subdirectory
try:
    os.mkdir(args.path+"\\ "+sub_dir_name_str)
except OSError as error:
    print(error)

try:
    os.mkdir(args.path + "\\ " + sub_dir_name_str+ "\\ " + "tropo")
except OSError as error:
    print(error)

for num in file_numbers:
    img_data = requests.get(url=image_url+str(num).zfill(3)+".png",
headers=headers).content
    with open(args.path + "\\ " + sub_dir_name_str+ "\\ " + "tropo" "\\ " +
str(num-12).zfill(2)+"_00_eeu.png", 'wb') as handler:
        handler.write(img_data)

```

					171.6321М.КР.ПЗ.Додатки	Арк.
						97
Змн.	Арк.	№ документа	Підпис	Дата		

ДОДАТОК Б

Код програми Create_geotiff

```
import os
import rasterio
from rasterio.transform import from_bounds
from PIL import Image
import numpy as np

# Шляхи до вхідної та вихідної директорій
input_directory = r""
output_directory = r""

# Географічні координати
top_left_geo = (71.42322798, 13.05194805)
top_right_geo = (71.42322798, 67.85714286)
bottom_left_geo = (39.63084071, 13.05194805)
bottom_right_geo = (39.63084071, 67.85714286)

# Піксельні координати зображення
top_left_pixel = (43, 66)
top_right_pixel = (887, 66)
bottom_left_pixel = (43, 654)
bottom_right_pixel = (887, 654)

# Обрізаємо зображення до розміру 845x589
crop_width, crop_height = 845, 589

# Процедура для обробки одного зображення
def process_image(image_path, output_path, transform):
    # Відкриваємо зображення
    image = Image.open(image_path)
    image = image.convert("RGB")

    # Обрізаємо зображення
    left_crop = (image.width - crop_width) // 2
    top_crop = (image.height - crop_height) // 2
    right_crop = left_crop + crop_width
    bottom_crop = top_crop + crop_height
    image = image.crop((left_crop, top_crop, right_crop, bottom_crop))

    # Перетворюємо зображення у масив
    image_data = np.array(image)
    width, height = image.size

    # Створюємо GeoTIFF
    with rasterio.open(
        output_path,
        'w',
        driver='GTiff',
        height=height,
        width=width,
        count=3,
        dtype=image_data.dtype,
        crs="EPSG:4326",
        transform=transform,
    ) as dst:
        dst.write(image_data[:, :, 0], 1) # Червоний канал
        dst.write(image_data[:, :, 1], 2) # Зелений канал
```

					171.6321М.КР.ПЗ.Додатки	Арк.
						98
Змн.	Арк.	№ документа	Підпис	Дата		

```

dst.write(image_data[:, :, 2], 3) # Синій канал

# Основна функція для обробки всіх зображень
def process_all_images(input_dir, output_dir):
    for date_folder in os.listdir(input_dir):
        date_folder_path = os.path.join(input_dir, date_folder)
        tropo_folder_path = os.path.join(date_folder_path, 'tropo')

        if os.path.isdir(tropo_folder_path):
            for image_file in os.listdir(tropo_folder_path):
                if image_file.endswith('_eeu.png'):
                    image_path = os.path.join(tropo_folder_path, image_file)

                    # Вихідна папка
                    output_date_folder = os.path.join(output_dir,
date_folder)

                    os.makedirs(output_date_folder, exist_ok=True)
                    output_image_path = os.path.join(output_date_folder,
image_file.replace('.png', '.tiff'))

                    # Географічні межі
                    left, right = top_left_geo[1], top_right_geo[1]
                    top, bottom = top_left_geo[0], bottom_left_geo[0]

                    # Створюємо affine-трансформацію
                    transform = from_bounds(left, bottom, right, top,
crop_width, crop_height)

                    # Обробляємо зображення
                    process_image(image_path, output_image_path, transform)

# Запускаємо процес
process_all_images(input_directory, output_directory)

```

					171.6321М.КР.ПЗ.Додатки	Арк.
						99
Змн.	Арк.	№ документу	Підпис	Дата		

ДОДАТОК В

Код програми getTropoData

```
import rasterio
import numpy
import numpy as np

# Цільовий колір (наприклад, [R, G, B])
target_color = {
    'Marginal': numpy.array([130, 0, 220]),
    'Fair': numpy.array([51, 119, 255]),
    'Moderate': numpy.array([2, 208, 161]),
    'High': numpy.array([160, 230, 50]),
    'Strong': numpy.array([230, 220, 50]),
    'Very_Strong': numpy.array([230, 175, 45]),
    'Intense': numpy.array([240, 130, 40]),
    'Very_intense': numpy.array([250, 60, 60]),
    'Extreme': numpy.array([255, 128, 192]),
    'Extreme+': numpy.array([255, 180, 220])
}

# Функція для отримання всіх унікальних кольорів
def get_unique_colors(file_path):
    # Відкриваємо GeoTIFF файл
    with rasterio.open(file_path) as dataset:
        # Читаємо всі канали (R, G, B) як окремі масиви
        red = dataset.read(1)
        green = dataset.read(2)
        blue = dataset.read(3)

        # Створюємо масив кольорів пікселів (R, G, B)
        colors = np.stack((red, green, blue), axis=-1)

        # Знаходимо всі унікальні комбінації кольорів
        unique_colors = np.unique(colors.reshape(-1, 3), axis=0)

    return unique_colors

def get_pixel_coords_by_color(file_path, target_color):
    """
    Повертає масив географічних координат пікселів з відповідним кольором.

    :param file_path: Шлях до GeoTIFF-файлу
    :param target_color: Цільовий колір у форматі [R, G, B]
    :return: Масив географічних координат або повідомлення про відсутність
    кольору
    """
    with rasterio.open(file_path) as src:
        # Читання даних зображення
        image_data = src.read() # image_data має форму (channels, height,
width)

        # Створюємо маску для пікселів з обраним кольором
        mask = np.all(image_data[:, 3] == target_color[:, None, None], axis=0)

        # Отримуємо індекси пікселів з відповідним кольором
        pixel_indices = np.argwhere(mask)

        if pixel_indices.size == 0: # Перевірка на відсутність пікселів з даним
```

					171.6321М.КР.ПЗ.Додатки	Арк.
						100
Змн.	Арк.	№ документа	Підпис	Дата		

```

кольором
    print(f"Колір {target_color} не знайдено у зображенні.")
    return []

    # Перетворення індексів пікселів у географічні координати, змінюємо
    порядок
    pixel_coords = [(src.xy(row, col)[1], src.xy(row, col)[0]) for row, col
in pixel_indices] # (широта, довгота) → (довгота, широта)

    return pixel_coords

def find_matching_colors_and_coords(file_path, target_colors=target_color):
    # Отримуємо всі унікальні кольори на зображенні
    unique_colors = get_unique_colors(file_path)

    # Словник для зберігання результатів
    matching_coords = {}

    # Проходимо по всіх кольорах, що були задані
    with rasterio.open(file_path) as src:
        for label, target_color in target_colors.items():
            if any(np.array_equal(target_color, uc) for uc in unique_colors):
                print(f"Знайдено колір {label}: {target_color}. Пошук
координат...")
                # Отримуємо координати для кожного знайденого кольору
                coords = get_pixel_coords_by_color(file_path, target_color)
                matching_coords[label] = coords
            else:
                print(f"Колір {label}: {target_color} не знайдено.")

    return matching_coords

def save_coords_with_colors(file_path, target_colors=target_color):
    matching_coords = find_matching_colors_and_coords(file_path,
target_colors)
    results = []

    for label, coords in matching_coords.items():
        for lat, lon in coords:
            results.append([lat, lon, label]) # Додаємо широту, довготу і
назву кольору

    return np.array(results)

#код для перевірок і тестів
path = r''

colors = get_unique_colors(path)
for color in colors:
    print(color)

coordinates_with_colors = save_coords_with_colors(path)
np.savetxt("coordinates_with_colors.csv", coordinates_with_colors,
delimiter=" ", fmt="%s")
print(coordinates_with_colors)

```

					171.6321М.КР.ПЗ.Додатки	Арк.
						101
Змн.	Арк.	№ документа	Підпис	Дата		

ДОДАТОК Г

Код програми для створення мапи тропосферного індексу

```
import os
from eomaps import Maps
import numpy as np
import matplotlib.pyplot as plt
import rasterio
from matplotlib.colors import to_hex
from Scripts.GetTropoData import find_matching_colors_and_coords

# Дані кольорів
target_color = {
    'Marginal': to_hex((130/255, 0/255, 220/255)),
    'Fair': to_hex((51/255, 119/255, 255/255)),
    'Moderate': to_hex((2/255, 208/255, 161/255)),
    'High': to_hex((160/255, 230/255, 50/255)),
    'Strong': to_hex((230/255, 220/255, 50/255)),
    'Very Strong': to_hex((230/255, 175/255, 45/255)),
    'Intense': to_hex((240/255, 130/255, 12/255)),
    'Very_intense': to_hex((250/255, 60/255, 60/255)),
    'Extreme': to_hex((255/255, 128/255, 192/255)),
    'Extreme+': to_hex((255/255, 180/255, 220/255))
}

# Шлях до головної папки
"""
base_directory = ''
"""
base_directory = r''
# Координати станції (LOT)
station_coords = (27.7, 42.43)
# Розміри мапи
Map_size = [26.06, 43.18, 40.02, 47.92]

# Функція для обробки кожного файлу та створення мап
def process_file(file_path, output_directory, tiff_filename):
    # Ініціалізуємо карту
    m = Maps(facecolor="none", figsize=(6, 3.5))

    # Додаємо елементи карти
    m.add_feature.preset.coastline()
    m.add_feature.preset.land()
    m.add_feature.preset.ocean()
    m.add_feature.preset.countries()

    # Викликаємо функцію для пошуку кольорів і координат
    matched_colors_coords = find_matching_colors_and_coords(file_path)

    # Виводимо результати
    for color_label, coords in matched_colors_coords.items():
        if coords:
            latitudes, longitudes = zip(*coords)
            m.ax.scatter(longitudes, latitudes, marker='.',
                color=target_color[color_label], label=str(color_label))

    # Налаштовуємо відображення карти
    m.ax.set_extent(Map_size)
    m.ax.scatter(*station_coords, color='blue', marker='x', s=100,
        label='Station LOT')
```

					171.6321М.КР.ПЗ.Додатки	Арк.
						102
Змн.	Арк.	№ документа	Підпис	Дата		

```

# Зберігаємо мапу в потрібну папку з відповідною назвою
fig = m.ax.get_figure()
output_filepath = os.path.join(output_directory, f"{tiff_filename}.png")
fig.savefig(output_filepath, dpi=300)
plt.close(fig) # Закриваємо фігуру, щоб уникнути витоку пам'яті

# Основний код для перебору папок і файлів
for folder_name in os.listdir(base_directory):
    folder_path = os.path.join(base_directory, folder_name)

    # Перевіряємо, чи це дійсно папка
    if os.path.isdir(folder_path):
        # Створюємо підпапку "Input" у цій папці
        input_folder = os.path.join(folder_path, 'Input')
        os.makedirs(input_folder, exist_ok=True)

        # Перебираємо всі файли в папці (тільки файли .tiff)
        for file_name in os.listdir(folder_path):
            if file_name.endswith('.tiff'):
                file_path = os.path.join(folder_path, file_name)

                # Назва TIFF-файла без розширення
                tiff_filename = os.path.splitext(file_name)[0]

                # Обробляємо кожен файл і зберігаємо карту
                process_file(file_path, input_folder, tiff_filename)

print("Обробка завершена!")

```

					171.6321М.КР.ПЗ.Додатки	Арк.
						103
Змн.	Арк.	№ документа	Підпис	Дата		

ДОДАТОК І

Код програми Station_Area_creator.py

```
import os
from eomaps import Maps
import numpy as np
from matplotlib.colors import to_hex
from matplotlib.patches import Polygon
from scipy.spatial import ConvexHull
from Scripts.GetTropoData import find_matching_colors_and_coords
import json
import matplotlib.pyplot as plt # For saving the figure

tiff_dir = ''
json_dir = r''

#номер станції
station_number = "12240"

#розміри мапи
Map_size = [26.06, 43.18, 40.02, 47.92]

#координати станції
station_coords = (40.21, 40.92)

# Дані кольорів
target_color = {
    'Marginal': to_hex((130 / 255, 0 / 255, 220 / 255)),
    'Fair': to_hex((51 / 255, 119 / 255, 255 / 255)),
    'Moderate': to_hex((2 / 255, 208 / 255, 161 / 255)),
    'High': to_hex((160 / 255, 230 / 255, 50 / 255)),
    'Strong': to_hex((230 / 255, 220 / 255, 50 / 255)),
    'Very_Strong': to_hex((230 / 255, 175 / 255, 45 / 255)),
    'Intense': to_hex((240 / 255, 130 / 255, 12 / 255)),
    'Very_intense': to_hex((250 / 255, 60 / 255, 60 / 255)),
    'Extreme': to_hex((255 / 255, 128 / 255, 192 / 255)),
    'Extreme+': to_hex((255 / 255, 180 / 255, 220 / 255))
}

def process_folders(tiff_dir, json_dir):
    # Проходимо через кожну папку з GeoTIFF файлами
    for root, dirs, files in os.walk(tiff_dir):
        for file in files:
            if file.endswith('.tiff'):
                # Отримуємо шлях до GeoTIFF файлу
                geotiff_path = os.path.join(root, file)

                # Генеруємо відповідний шлях до JSON файлу
                date_folder_name = os.path.basename(root) # Отримуємо назву
                папки (наприклад, 01_04_2023)

                # Шукаємо відповідний JSON файл (припускаємо, що назви
                збігаються)
                json_folder = os.path.join(json_dir, date_folder_name,
                station_number)

                # Перевіряємо наявність відповідного JSON файла
                json_filename = file.replace('eeu.tiff', 'map.json') #
```

					171.6321М.КР.ПЗ.Додатки	Арк.
						104
Змн.	Арк.	№ документа	Підпис	Дата		

```

Міняємо суфікс
json_path = os.path.join(json_folder, json_filename)

if not os.path.exists(json_path):
    print(f"Попередження: Не знайдено відповідного JSON для
{geotiff_path}. Пропускаю.")
    continue # Пропускаємо цей GeoTIFF, якщо немає
відповідного JSON

# Створюємо вихідну папку для збереження мап
output_subfolder = os.path.join(root, 'Output')
os.makedirs(output_subfolder, exist_ok=True)

# Обробляємо кожен GeoTIFF файл з відповідним JSON файлом
process_geotiff_and_json(geotiff_path, json_path,
output_subfolder)

def process_geotiff_and_json(geotiff_path, json_path, output_dir,
station_coords=station_coords):
    # Ініціалізуємо карту
    m = Maps(facecolor="none", figsize=(6, 3.5))

    # Додаємо елементи карти
    m.add_feature.preset.coastline()
    m.add_feature.preset.land()
    m.add_feature.preset.ocean()
    m.add_feature.preset.countries()

    # Завантажуємо кольори з тега TIFF
    matched_colors_coords = find_matching_colors_and_coords(geotiff_path)

    # Виводимо результат на мапу
    for color_label, coords in matched_colors_coords.items():
        if coords:
            latitudes, longitudes = zip(*coords)
            m.ax.scatter(longitudes, latitudes, marker='.',
color=target_color[color_label], label=str(color_label))

    # Парсимо дані JSON
    with open(json_path, 'r') as f:
        coordinates = json.load(f)

    # Створюємо список координат для полігону
    polygon_coords = [(float(item["LON"]), float(item["LAT"])) for item in
coordinates]

    # Якщо координати порожні, вивести попередження
    if not polygon_coords:
        print(f"Попередження: У JSON файлі {json_path} немає координат.")
        return

    # Обчислюємо опуклу оболонку
    points = np.array(polygon_coords)
    hull = ConvexHull(points)

    # Додаємо полігон покриття на мапу (повністю червоний і без прозорості)
    hull_points = points[hull.vertices]
    polygon = Polygon(hull_points, closed=True, edgecolor='red',
facecolor='red', fill=True, linewidth=2, zorder=10)
    m.ax.add_patch(polygon)

    # Додаємо координати станції

```

					171.6321М.КР.ПЗ.Додатки	Арк.
						105
Змн.	Арк.	№ документа	Підпис	Дата		

```

m.ax.scatter(*station_coords, color='red', marker='x', s=100,
label='Station LOT', zorder=11)

# Налаштовуємо кордони карти
m.ax.set_extent(Map_size)

# Зберігаємо мапу з тим самим ім'ям, що й у TIFF файлу
output_filename = os.path.join(output_dir,
os.path.basename(geotiff_path).replace('.tiff', '.png'))

# Зберігаємо фігуру
plt.savefig(output_filename, dpi=300)
print(f"Мапу збережено як {output_filename}")

# Закриваємо фігуру, щоб уникнути проблем з пам'яттю
plt.close()

# Запускаємо процес обробки
process_folders(tiff_dir, json_dir)

```

					171.6321М.КР.ПЗ.Додатки	Арк.
						106
Змн.	Арк.	№ документу	Підпис	Дата		

ДОДАТОК Д

Код програми для створення бінарних масок

```
import cv2
import numpy as np
import os

def process_images_in_folder(input_folder):
    # Отримати список всіх файлів у папці
    for filename in os.listdir(input_folder):
        # Перевірка чи файл є зображенням (перевіряємо розширення)
        if filename.endswith(".png") or filename.endswith(".jpg") or
filename.endswith(".jpeg"):
            file_path = os.path.join(input_folder, filename)

            # Читання зображення з полігоном
            polygon_image = cv2.imread(file_path)

            if polygon_image is None:
                print(f"Не вдалось відкрити зображення {filename}")
                continue

            # Перетворення зображення в HSV-простір для виділення
червоного кольору
            hsv = cv2.cvtColor(polygon_image, cv2.COLOR_BGR2HSV)

            # Діапазон кольорів для червоного кольору
            lower_red1 = np.array([0, 70, 50])
            upper_red1 = np.array([10, 255, 255])
            lower_red2 = np.array([170, 70, 50])
            upper_red2 = np.array([180, 255, 255])

            # Маска для виділення червоного кольору
            mask1 = cv2.inRange(hsv, lower_red1, upper_red1)
            mask2 = cv2.inRange(hsv, lower_red2, upper_red2)
            mask = mask1 | mask2

            # Перетворення маски в бінарне зображення (0 або 255)
            binary_mask = cv2.threshold(mask, 127, 255,
cv2.THRESH_BINARY)[1]

            # Збереження маски на місці оригінального зображення
            cv2.imwrite(file_path, binary_mask)
            print(f"Оброблено зображення {filename}")

# Вкажіть шлях до папки з вашими зображеннями
input_folder = '/content/drive/MyDrive/Diplom2024/Тесты/Output'
process_images_in_folder(input_folder)
```

					171.6321М.КР.ПЗ.Додатки	Арк.
						107
Змн.	Арк.	№ документа	Підпис	Дата		

ДОДАТОК Е

Код для тренування нейромережевої моделі

```
import os
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.preprocessing.image import load_img, img_to_array
from sklearn.model_selection import train_test_split

# Шлях до папок з даними
input_dir = '' # Папка з вхідними зображеннями мап
output_dir = '' # Папка з масками покриття

# Розмір зображень для вхідних даних і масок
IMG_HEIGHT, IMG_WIDTH = 256, 256

# Функція для завантаження зображень
def load_images(image_dir, mask_dir, img_height, img_width):
    images = []
    masks = []

    image_files = sorted(os.listdir(image_dir))
    mask_files = sorted(os.listdir(mask_dir))

    for img_file, mask_file in zip(image_files, mask_files):
        img_path = os.path.join(image_dir, img_file)
        mask_path = os.path.join(mask_dir, mask_file)

        # Завантаження зображення та маски
        img = load_img(img_path, target_size=(img_height, img_width))
        mask = load_img(mask_path, target_size=(img_height, img_width),
            color_mode="grayscale")

        img = img_to_array(img) / 255.0 # Нормалізуємо зображення до [0,
1]
        mask = img_to_array(mask) / 255.0 # Маски також в межах [0, 1]

        images.append(img)
        masks.append(mask)

    return np.array(images), np.array(masks)

# Завантажуємо зображення та маски
images, masks = load_images(input_dir, output_dir, IMG_HEIGHT, IMG_WIDTH)

# Розділяємо дані на тренувальні та тестові
```

					171.6321М.КР.ПЗ.Додатки	Арк.
						108
Змн.	Арк.	№ документа	Підпис	Дата		

```

X_train, X_test, y_train, y_test = train_test_split(images, masks,
test_size=0.2, random_state=42)

# Створюємо модель DeepLabv3+
def create_deeplabv3plus_model(input_shape):
    # Використовуємо базову модель DenseNet121 без верхніх шарів
    base_model =
tf.keras.applications.DenseNet121(input_shape=input_shape,
include_top=False)
    base_model.trainable = False

    # Вхідний шар
    inputs = keras.Input(shape=input_shape)

    # Основна модель DeepLabv3+
    x = base_model(inputs, training=False)

    # Сегментаційні шари
    x = layers.Conv2D(256, 3, padding="same", activation="relu")(x)
    x = layers.Conv2D(128, 3, padding="same", activation="relu")(x)
    x = layers.Conv2D(64, 3, padding="same", activation="relu")(x)

    # Upsampling для відновлення початкової розмірності (256x256)
    x = layers.UpSampling2D(size=(4, 4), interpolation='bilinear')(x) #
Збільшуємо до (32x32)
    x = layers.Conv2D(32, 3, padding="same", activation="relu")(x)
    x = layers.UpSampling2D(size=(8, 8), interpolation='bilinear')(x) #
Збільшуємо до (256x256)

    # Вихідний шар - прогноз маски (форма має бути (256, 256, 1))
    outputs = layers.Conv2D(1, 1, activation="sigmoid")(x)

    # Створюємо модель
    model = keras.Model(inputs, outputs)
    return model

# Створюємо модель
model = create_deeplabv3plus_model((IMG_HEIGHT, IMG_WIDTH, 3))

# Компілюємо модель
model.compile(optimizer="adam", loss="binary_crossentropy",
metrics=["accuracy"])

# Навчаємо модель
history = model.fit(X_train, y_train, epochs=60, batch_size=8,
validation_data=(X_test, y_test))

# Оцінюємо модель
model.evaluate(X_test, y_test)

```

					171.6321М.КР.ПЗ.Додатки	Арк.
						109
Змн.	Арк.	№ документа	Підпис	Дата		

```
# Збереження моделі у форматі Keras
model.save('')

# Або збереження у форматі HDF5
model.save('')
```

					171.6321М.КР.ПЗ.Додатки	Арк.
						110
Змн.	Арк.	№ документа	Підпис	Дата		

ДОДАТОК Є

Код для оцінки прогнозів нейромережі

```
import cv2
import numpy as np
import os
import matplotlib.pyplot as plt

def calculate_polygon_correlation(image_path1, image_path2):
    # Завантаження зображень
    image1 = cv2.imread(image_path1, cv2.IMREAD_GRAYSCALE)
    image2 = cv2.imread(image_path2, cv2.IMREAD_GRAYSCALE)

    # Бінаризація зображень (чорне -> 0, біле -> 255)
    _, binary1 = cv2.threshold(image1, 127, 255, cv2.THRESH_BINARY)
    _, binary2 = cv2.threshold(image2, 127, 255, cv2.THRESH_BINARY)

    # Перевірка чи розміри зображень однакові
    if binary1.shape != binary2.shape:
        raise ValueError(f"Зображення {image_path1} і {image_path2} мають різні розміри!")

    # Обчислення кореляції між двома бінарними зображеннями
    correlation = np.corrcorr(binary1.flatten(), binary2.flatten())[0, 1]

    # Конвертація ступені збігу в відсотки
    correlation_percent = correlation * 100

    return correlation_percent

def process_images(folder1, folder2):
    # Отримати список файлів у кожній папці
    files1 = set(os.listdir(folder1))
    files2 = set(os.listdir(folder2))

    # Знайти спільні файли в обох папках
    common_files = files1.intersection(files2)

    correlations = []
    file_names = []

    for file_name in common_files:
        image_path1 = os.path.join(folder1, file_name)
        image_path2 = os.path.join(folder2, file_name)

        try:
            correlation = calculate_polygon_correlation(image_path1,
image_path2)
```

					171.6321М.КР.ПЗ.Додатки	Арк.
						111
Змн.	Арк.	№ документа	Підпис	Дата		

```

        correlations.append(correlation)
        file_names.append(file_name)
        print(f"{file_name}: ступінь збігу {correlation:.2f}%")
    except ValueError as e:
        print(e)

    # Обчислення середньої ступені збігу
    average_correlation = np.mean(correlations) if correlations else 0
    print(f"\nСередня ступінь збігу для всіх прогнозів:
{average_correlation:.2f}%")

    return file_names, correlations, average_correlation

def plot_correlation(file_names, correlations, average_correlation):
    plt.figure(figsize=(12, 6))
    plt.bar(file_names, correlations, color='skyblue')
    plt.xticks(rotation=90)
    plt.xlabel('Назва файлу')
    plt.ylabel('ступінь збігу (%)')
    plt.title('ступінь збігу покриття між marinetraffic та прогнозом
мережі')

    # Додаємо середню кореляцію як горизонтальну лінію
    plt.axhline(y=average_correlation, color='red', linestyle='--',
label=f'Середня ступінь збігу: {average_correlation:.2f}%')

    # Більш детальна шкала на осі Y з кроком 10
    plt.yticks(np.arange(0, 101, 10))

    plt.legend()
    plt.tight_layout()
    plt.show()

# Використання
folder1 = ''
folder2 = ' '

file_names, correlations, average_correlation = process_images(folder1,
folder2)
plot_correlation(file_names, correlations, average_correlation)

```

					171.6321М.КР.ПЗ.Додатки	Арк.
						112
Змн.	Арк.	№ документа	Підпис	Дата		

ДОДАТОК Ж

Код для побудови моделі лінійної регресії

```
import json
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score

# Завантаження даних з JSON-файлу
input_file = "/content/drive/MyDrive/Diplom2024/Vovka.json"
with open(input_file, "r", encoding="utf-8") as f:
    data = json.load(f)

# Підготовка даних для моделі
dates = []
areas = []
signal_counts = []

for day, values in data.items():
    # Беремо перше значення з Area і AisSignals (якщо вони списки)
    area = values["Area"][0] if values["Area"] else 0
    signals = values["AisSignals"][0] if values["AisSignals"] else 0

    # Додаємо до списків тільки якщо є дані та кількість сигналів менше
    # або дорівнює 150
    if area > 0 and signals > 0 and signals <= 150:
        dates.append(day)
        areas.append(area)
        signal_counts.append(signals)

# Конвертуємо у масиви NumPy
X = np.array(areas).reshape(-1, 1) # Площа (вектор вхідних даних)
y = np.array(signal_counts)        # Кількість сигналів (вихідні дані)

# Розділення на навчальну і тестову вибірки
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
                                                    random_state=42)

# Створення і навчання моделі лінійної регресії з нульовим вільним членом
model = LinearRegression(fit_intercept=False)
model.fit(X_train, y_train)

# Прогнозування
y_pred = model.predict(X_test)

# Оцінка моделі
```

					171.6321М.КР.ПЗ.Додатки	Арк.
						113
Змн.	Арк.	№ документа	Підпис	Дата		

```

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse:.2f}")
print(f"R2 Score: {r2:.2f}")

# Графік залежності площі і кількості сигналів
plt.figure(figsize=(8, 5))
plt.scatter(X, y, color="blue", label="Дані")
plt.plot(X, model.predict(X), color="red", label="Лінія регресії")
plt.xlabel("Площа (Area)")
plt.ylabel("Кількість сигналів (AisSignals)")
plt.title("Залежність між площею і кількістю сигналів")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

# Наприклад, площа 120
area_to_predict = np.array([[2581]]) # Перетворення на 2D-масив
predicted_signals = model.predict(area_to_predict)

print(f"Прогнозована кількість сигналів для площі {area_to_predict[0][0]}: {predicted_signals[0]:.2f}")

```

					171.6321М.КР.ПЗ.Додатки	Арк.
						114
Змн.	Арк.	№ документа	Підпис	Дата		