

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра програмного забезпечення

## **Пояснювальна записка**

до магістерської кваліфікаційної роботи

магістр

(освітньо-кваліфікаційний рівень)

на тему «Розробка програмних методів оптимізації комп'ютерної системи прогнозування рівня води річок Прикарпаття»

Виконав: студент II курсу, групи 1ПІ-18м  
напряму підготовки  
121 – Інженерія програмного забезпечення  
(шифр і назва напряму підготовки, спеціальності)

Гусак М.А

(прізвище та ініціали)

Керівник Петух А.М., Романюк О.Н.

(прізвище та ініціали)

Рецензент Колодний В.В.

(прізвище та ініціали)

ВНТУ – 2019 року

Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра програмного забезпечення  
Освітньо-кваліфікаційний рівень – магістр  
Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ

Романюк О.Н.

“ \_\_\_ ” \_\_\_\_\_ 2019 року

## **З А В Д А Н Н Я НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Гусаку Михайлу Анатолійовичу

1. Тема роботи– розробка програмних методів оптимізації комп'ютерної системи прогнозування рівня води річок Прикарпаття.

Керівник роботи: Петух Анатолій Михайлович д.т.н., проф. кафедри ПЗ, консультант – Романюк Олександр Никифорович, д.т.н., зав. кафедри ПЗ, затверджені наказом вищого навчального закладу від “ \_\_\_ ” \_\_\_\_\_ 2019 року № \_\_\_\_\_

2. Строк подання студентом роботи

---

3. Вихідні дані до роботи: тип річок– річки Прикарпаття; алгоритм опису процесів з некротними частотами–на основі генетичних алгоритмів; прогнозування стану річки – за спостереженнями; поточні дані про параметри погодних умов: температура, вологість, атмосферний тиск, опади, напрямок вітру та дані рівня води; мова програмування - C#.

4. Зміст розрахунково-пояснювальної записки: вступ; аналіз методів оптимізації комп'ютерної системи прогнозування рівня води річок; розробка математичної моделі на основі генетичних алгоритмів; розробка програмного методу оптимізації комп'ютерної системи прогнозування рівня води річок; економічна частина; висновки; перелік посилань.

5. Перелік графічного матеріалу: тема; актуальність; мета, об'єкт та предмет дослідження; наукова новизна отриманих результатів; практична цінність; математична модель екологічної поведінки річки; метод оптимізації прогнозування рівня води; складові генетичного алгоритму; комп'ютерна система прогнозування рівня води; схема трафіку даних; висновки.

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав(-ла)	завдання прийняв
1-4	Романюк О. Н., д.т.н., зав. кафедри ПЗ		
5	Бальзан М.В., к.е.н., доц. кафедри ЕПВМ		

7. Дата видачі завдання \_\_\_\_\_

**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів магістерської кваліфікаційної Роботи	Строк виконання етапів роботи	Примітка
1	Аналіз існуючих методів оптимізації комп'ютерної системи прогнозування рівня води річок	23.09.2019 – 03.10.2019	Вик.
2	Розробка математичної моделі на основі генетичних алгоритмів	03.10.2019 – 21.10.2019	Вик.
3	Розробка програмного методу оптимізації комп'ютерної системи прогнозування рівня води річок	21.10.2019 – 10.11.2019	Вик.
4	Економічна частина	10.11.2019 – 24.11.2019	Вик.

Студент \_\_\_\_\_ Гусак М.А.  
 ( підпис ) ( прізвище та ініціали )

Керівник магістерської кваліфікаційної роботи \_\_\_\_\_ Романюк О.Н.  
 ( підпис ) ( прізвище та ініціали )

## АННОТАЦІЯ

У магістерській кваліфікаційній роботі, проведено детальний аналіз методів і засобів моніторингу та прогнозування рівня води річок. Сформульовано мету досліджень - підвищення рівня прогнозування рівня води річок за рахунок розробки моделі та методу оптимізації.

Запропоновано математичну модель екологічної поведінки річки, у якій, на відміну від відомої, враховано коливальний процес та вагові коефіцієнти гармонійних складових, що дозволило підвищити точність моделі а також, метод оптимізації прогнозування рівня води річок, який відрізняється від відомих використанням нейромереж, що дозволило підвищити достовірність оптимізації та комп'ютеризувати його.

Розроблено програмний додаток для підвищеної точності прогнозування рівня води річок.

Отримані в магістерській кваліфікаційній роботі наукові та практичні результати можна використати для прогнозування рівня води річок.

## ABSTRACT

In the master's qualification work, a detailed analysis of methods and means of monitoring and forecasting the water level of rivers was carried out. The research goal is formulated to increase the level of river water forecasting by developing a model and an optimization method.

A mathematical model of ecological behavior of a river is proposed, which, unlike the known one, takes into account the oscillation process and the weight coefficients of the harmonic components, which allowed to improve the accuracy of the model, as well as a method of optimizing the prediction of the water level of the rivers, which differs from the known ones using the unbelief optimize and computerize it.

A software application has been developed to improve the accuracy of river water level forecasting.

The scientific and practical results obtained in the master's qualification work can be used to predict the water level of rivers.

## ЗМІСТ

1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ОПТИМІЗАЦІЇ КОМП'ЮТЕРНОЇ СИСТЕМИ ПРОГНОЗУВАННЯ РІВНЯ ВОДИ РІЧОК.....	10
1.1. Аналіз методів і засобів моніторингу і прогнозування рівня води річок.....	10
1.2. Технічні засоби вимірювання метрологічних параметрів.....	<b>Ошибка!</b>
<b>Закладка не определена.</b>	
1.3. Вибір і обґрунтування напрямку досліджень.....	17
1.4. Висновки до розділу.....	18
2 РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ НА ОСНОВІ ГЕНЕТИЧНИХ АЛГОРИТМІВ.....	3
2.1. Алгоритм опису процесів з некротними частотами на основі генетичних алгоритмів.....	3
2.2. Складові генетичного алгоритму.....	28
2.3. Висновки до розділу.....	32
3 РОЗРОБКА ПРОГРАМНОГО МЕТОДУ ОПТИМІЗАЦІЇ КОМП'ЮТЕРНОЇ СИСТЕМИ ПРОГНОЗУВАННЯ РІВНЯ ВОДИ РІЧОК.....	3
3.1. Синтез структури комп'ютерної системи моніторингу і прогнозування рівня води в річці.....	3
3.2. Обґрунтування і вибір операційної системи програмного забезпечення.....	3
3.3. Вибір технічних засобів моніторингу та прогнозування рівня води в річках.....	3
3.4. Висновки до розділу.....	51
4 ЕКОНОМІЧНА ЧАСТИНА.....	3
4.1. Оцінювання комерційного потенціалу розробки.....	3
4.2. Прогнозування витрат на виконання наукової роботи та впровадження її результатів.....	3
4.3. Прогнозування комерційних ефектів від реалізації результатів розробки.....	3
4.4. Розрахунок ефективності вкладених інвестицій та періоду їх окупності.....	3
4.5. Висновки до розділу.....	65
ВИСНОВКИ.....	67
ПЕРЕЛІК ПОСИЛАНЬ.....	68
Додаток А. Технічне завдання.....	73
Додаток Б. Ілюстративний матеріал.....	77
Додаток В. Лістинг головного модуля програми.....	91

## ВСТУП

Причини виникнення паводків можна поділити на дві великі групи: природні та антропогенні.

Можна вважати, що атмосферні опади та танення снігів – це природні причини виникнення паводкових вод. Але, формування вод починається тоді, коли опади перевищують 20 мм на добу. При дуже інтенсивних зливах, приблизно 100 мм опадів, паводки можуть нести за собою руйнівний характер. Найчастіше рівень снігу в Прикарпатті досягає 0,6 м, але цей показник є нестабільним і при певних умовах може сягати навіть до 2 метрів, що в свою чергу при швидкому таненні спричиняє стрімке та катастрофічне підняття води.

До антропогенних причин виникнення паводків відносять:

- руйнування природних стоків;
- вирубування лісів;
- розорювання схилів.

На руйнування природних стоків впливає як видобуток піску та гравію в річках, а також штучне осушування територій. Зловживання, розорювання схилів може спричинити не достатню затримку дощових та талих вод. Стрімке вирубування лісів значно зменшує максимально можливий вологовміст території, що в свою чергу спричиняє розвиток паводків на території водозбору.

Існуючі методи контролю рівня вод, які використовують програмне і апаратне забезпечення, класифікуються як:

- ручні методи;
- автоматизовані методи контролю.

Для розробки програмних методів прогнозування рівня води річок на Прикарпатті необхідно підвищити достовірність прогнозування для зменшення негативних наслідків. Це передбачає розробку нових методів і засобів, орієнтованих на комп'ютерну обробку.

**Зв'язок роботи з науковими програмами, планами, темами.** Робота виконувалася згідно плану виконання наукових досліджень на кафедрі програмного забезпечення.

**Метою магістерської кваліфікаційної роботи** є підвищення рівня прогнозування рівня води річок за рахунок розробка моделі та методу оптимізації.

**Основними задачами дослідження є:**

- провести аналіз існуючих методів і засобів аналізу рівня води річок;
- розробка моделі екологічної поведінки річки, у якій на відміну від відомої враховано коливальний процес та вагові коефіцієнти гармонійних складових, що дозволить підвищити точність моделі;
- розробка методу оптимізації прогнозування рівня води річок Прикарпаття, який відрізняється від відомого використанням нейромереж, що дозволило підвищити достовірність оптимізації та комп'ютеризувати його;
- розробити алгоритми та програмні засоби на основі запропонованих методів та моделей;
- провести експериментальні дослідження розроблених засобів.

**Об'єкт дослідження** – процес прогнозування рівня води річок.

**Предмет дослідження** – методи та засоби прогнозування рівня води річок.

**Методи дослідження.** У процесі досліджень використовувались: теорія чисел та чисельних методів, теорія нейромереж, теорія гармонійного аналізу для розробки методу та моделі прогнозування; комп'ютерне моделювання для аналізу та перевірки отриманих теоретичних положень.

**Наукова новизна отриманих результатів.**

1. Вперше розроблено математичну модель екологічної поведінки річки, у якій, на відміну від відомої, враховано коливальний процес та вагові коефіцієнти гармонійних складових, що дозволило підвищити точність моделі.



2. Подальшого розвитку отримав метод оптимізації прогнозування рівня води річок, який відрізняється від відомого використанням нейромереж, що дозволило підвищити достовірність оптимізації та комп'ютеризувати його.

**Практична цінність отриманих результатів.** Практична цінність одержаних результатів полягає в тому, що на основі отриманих у магістерській кваліфікаційній роботі теоретичних положень запропоновано алгоритми та розроблено програмні засоби для прогнозування рівня річок.

**Особистий внесок здобувача.** Усі наукові результати, викладені у магістерській кваліфікаційній роботі, отримані автором особисто. У роботі [1] автору належать такі результати: вагові коефіцієнти гармонійних складових.

**Апробація матеріалів магістерської кваліфікаційної роботи.** Основні положення магістерської кваліфікаційної роботи доповідалися та обговорювалися на Міжнародній науково-практичній конференції «Електронні інформаційні ресурси: створення, використання, доступ» (Вінниця, 2019).

# 1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ОПТИМІЗАЦІЇ КОМП'ЮТЕРНОЇ СИСТЕМИ ПРОГНОЗУВАННЯ РІВНЯ ВОДИ РІЧОК

## 1.1 Аналіз методів і засобів моніторингу і прогнозування рівня води річок

Екосистеми, з точки зору системного аналізу вважаються складними системами, які змінюють стан з плином часу, що прогнозує їх майбутню поведінку при певних умовах. Є кілька підходів для прогнозування поведінки складних систем: використання інтуїції, експериментальні дослідження ідентичних або подібних системи, і, нарешті, математичне моделювання.

Всі ці підходи використовуються в екологічних дослідженнях. Проте, на сьогодні час характер і ступінь втручання людини в природні процеси величезний, і в більшості випадків, дослідницька інтуїція і досвід, не працюють. Зі зрозумілих причин, можливості провести експеримент над природними екосистемами дуже обмежені, і в більшості випадків банально неможливі.

Математичний опис поведінки організмів був створений як одна з перших робіт в області моделі екосистеми.

Розглядаючи стабільність екосистем, Дж Сміт використовує такі поняття, як  $n$ -мірні фазовий простір, фазові траєкторії, які в результаті змінного вектора руху стану екосистеми. Ці змінні «щільність видів», які утворюють відповідну екосистему.

Наприклад, описуючи модель взаємодії між типом щільності хижаків і щільністю жертви, яка знищена хижаком приходиться до такої системи диференціальних рівнянь:

$$\frac{dx}{dt} = -ex + dxy, \quad (1.1)$$

$$\frac{dy}{dt} = ay - by^2 - cxy, \quad (1.2)$$

де  $a$  - швидкість росту жертви;

$\frac{a}{b}$  - ємність середовища;

$c, d$  - постійні величини.

Для того, щоб оцінити поведінку системи, яка визначена за допомогою математичної моделі (1.1) і (1.2) була розгорнута в фазовій траєкторії. Для визначення умов, при яких екосистема знаходиться в рівновазі з (1.1) і (1.2) отримано систему рівнянь:

$$\begin{aligned} -e + dy &= 0, \\ a - by - cx &= 0, \end{aligned}$$

результат, який визначає координати точки рівноваги в фазовому просторі

$(x, y)$ , оскільки,  $y > 0$ , то  $x$  матиме позитивне значення в разі, коли  $\frac{a}{b} > \frac{e}{d}$ ;

Цей приклад ілюструє ідентичність динамічних властивостей, технічних систем і екосистем, які описані за допомогою детермінованих математичних моделей. Детерміновані моделі відтворюють поведінку складних екосистем лише частково.

Характеристикою складної системи «людина-біосфера» є стійкість цієї біосистеми. Це основна властивість всіх динамічних систем. Стійкість руху та стійкість динамічних процесів як поняття зародились і дістали розвиток стосовно скланих механічних процесів. Як виявилось, визначення стійкості автоматичних систем керування базувалось на отриманих результатах. На наш час теорія стійкості динамічних систем дуже популярна і використовується в різних галузях науки. Стійкість – це

властивість, яка залежна від початкових умов та інших властивостей системи[2].

Для опису динамічних властивостей об'єктів було обрано системи рівнянь, а для дослідження їх стійкості буде використано метод Ляпунова.

Перший метод Ляпунова був запропонований Т. Мальтусом у 1802 році, ця функція була названа в честь математика Олександра Ляпунова.

$$\frac{dx}{dt} = ax - bx^2; \quad x(t_0) = x_0, \quad (1.3)$$

де  $x(t)$  - число особин у популяції на момент часу  $t$ ;

$x_0$  - число особин у початковий момент часу  $t_0$ ;

$a, b$  - постійні величини, які характеризують інтенсивність розмноження і вимирання особин популяції.

Розв'язок рівняння (1.3) буде таким:

$$x(t) = \frac{\frac{a}{b}x_0}{x_0 + \left(\frac{a}{b} - x_0\right)e^{-a(t-t_0)}}.$$

Для екосистеми доступно два положення рівноваги, визначені із

умови  $\frac{dx_p}{dt} = 0$ , тобто  $x_p^{(1)} = 0$  і  $x_p^{(2)} = \frac{a}{b}$ .

При  $x_0 \neq 0$  з плином часу у системі встановлюється стан рівноваги

$x_p = \frac{a}{b}$ , який, згідно першого правила Ляпунова, є стійким: якщо популяція

більше цього значення  $(x_0 > x_p)/(x_0 > x_p)$ , то вона з плином часу зменшується, менше  $(x_0 < x_p)/(x_0 < x_p)$  – збільшується (рис. 1.1).

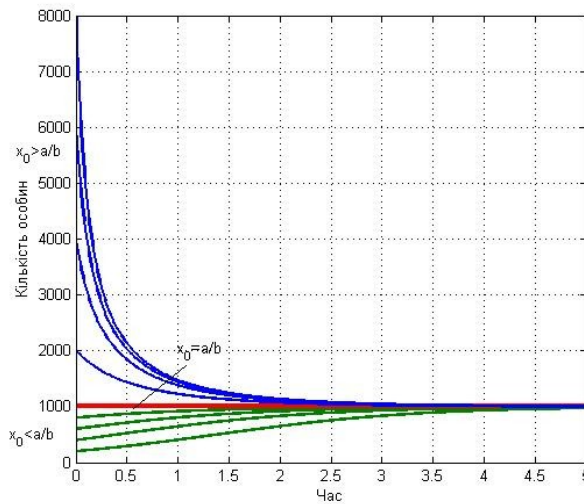


Рисунок 1.1. Динамічна зміна популяції у залежності від початкових умов ( $t_0 = 0$ )

Антропогенний вплив на середовище неможливо більше ігнорувати. Соціальний прогрес, який дав людству переваги сучасної цивілізації також має іншу сторону медалі, що все більше і більше проявляється в порушенні природних процесів екосистем[3].

Виробнича діяльність соціуму можна охарактеризувати явним збільшенням використання енергії та різних ресурсів, велика частина яких є вичерпною. Газ, нафта є джерелом енергії, яка виходить шляхом спалювання значної частини природного газу та нафтопродуктів. Деяка частина речовин, які беруть участь у виробничій діяльності людства, повертається в навколишнє середовище у вигляді продуктів згоряння, відходів, добрив, пестицидів і так далі. Таким чином, в двадцятому столітті людство зіштовхнулося з проблемою прогнозування наслідків антропогенного впливу на навколишнє середовище. У більшості випадків такий вплив є негативним, і може призвести до екологічних катастроф.

Сучасними математичними моделями екологічних процесів є – емпіричні, аналітичні, та імітаційні.

Аналітичні моделі побудовані для визначення внутрішніх властивостей екосистеми і його взаємодію з навколишнім середовищем. Ці моделі здатні відтворити явища, які можуть спостерігатися в поведінці екосистем, такі як взаємодія видів, кисневий цикл, коливальний характер змін в системі біомаси, освіту нерівностей в просторову структуру і так далі. Аналітичні моделі мають універсальні властивості. Якщо вона адекватно відображає математичну форму поведінки системи, ця модель може передбачити наслідок впливу людини на певних екосистемах і біосфері. Створення таких моделей є досить складною і проблематичною завданням, оскільки деякі екосистеми можуть взаємодіяти і ця взаємодія дуже часто має імовірнісний характер, яка робить мета побудови точних прогностичні моделям нездійсненним завдання в більшості випадків[4].

Отже, продуктивний підхід повинен бути заснований на обробці експериментальних даних. В результаті, він отримав емпіричну модель, яка належить до другого класу математичних моделей. Перевагою емпіричних моделей є надійність результатів. Недолік такої моделі – це обмежене значення отриманих результатів.

Імітаційній моделі та їх розвиток переплітаються з динамічним впровадженням у наукову діяльність ЕОМ.

Картографічне моделювання – це напрямок дослідження на основі якого створюють картографічні моделі, а також розробляють методи і прийоми їх аналізу.

Популярним методом моделювання також вважають картографічне моделювання геохімічних полів, аналіз цих полів допомагає спрогнозувати швидкість поширення хімічних елементів на конкретній території, яка було обрана під час картографічного моделювання[5]. В свою чергу, цей метод допомагає раціонально використовувати природні ресурси нашої планети.

Картографічне моделювання складається з таких основних етапів як:

- збір первинних даних;
- робота з укладання карт;

- картографічні моделі;
- кодування локації.

Як правило, для побудови таких моделей найкраще підходить інтерполяція.

## 1.2 Технічні засоби вимірювання метрологічних параметрів

До вивчення гідрологічного режиму рік України приступили приблизно у другій половині XIX століття.

Якщо характеризувати стан гідрологічної мережі України на сьогодні, слід відзначити стрімке скорочення гідрометеорологічних постів.

Для визначення гідрологічних характеристик застосовуються прості механічні пристрої, вони потребують ручної праці спостерігача, 10 відсотків припадає на механічні пристрої, за допомогою визначення характеристик є неефективним внаслідок значної розосередженості.

Основна і допоміжна програми використовуються для спостереження за станом різноманітних водойм, вони ведуться на постійних або тимчасових постах.

Передаточні водомірні пости споруджуються тоді, коли підхід до води важкодоступний (наприклад, дуже круті береги). Рівні води в таких випадках вимірюються вище поверхні води або збоку на березі. Передаточний водомірний пост обладнується як на березі, так і на гідротехнічній споруді. Мостовий водомірний пост облаштовується на містку чи іншій споруді, розташованій над водою. Для цього на мосту (чи іншій споруді) вибирається і закріплюється постійна точка (нуль спостережень), висотне положення якої визначається нівелюванням від репера. Рівень води вимірюється рейкою, розміченим тросом чи рулеткою з тягарцем[6].

Дистанційні водомірні складаються з датчика, джерела живлення, каналу зв'язку і реєстратора даних про висоту рівня. Більшість дистанційних рівнемірів мають датчик у вигляді поплавкового пристрою[7].

Останім часом було розроблено багато акустичних вимірювачів висоти рівня, принцип дії яких базується на проміжку часу за який проходить звуковий імпульс, відбитий від поверхні води. Вони встановлюються над поверхнею води, акустичний випромінювач надсилає звуковий імпульс в напрямку поверхні води. З свого боку, приймач чітко фіксує відбитий імпульс, і заміряє час, що знадобився для визначення відстані до водної поверхні[8]. Температури і вологість впливаються напряму на швидкість звуку в повітрі, він змінюється залежно від цих атрибутів. Вимірювання компенсують деякі постійні відстані і показники температури повітря в вимірювальній зоні[9]. Слід відзначити, що всі вимірювання виконуються у звуковій езоляційній трубі, що слугує для захисту від різких змін робочої зони вимірювача[10].

Пневматичні системи засновані на вимірюванні тиску стовпа води над обраною точкою і форматуванні його в еквівалентний рівень водойми. Пневматичні системи враховують щільність води і гравітаційну константу. Перетворювач тиску слугує для заміру рівня, який базується під найнижчим очікуваним рівнем. Найбільшим недоліком є низька працездатність в умовах посиленої дії хвиль[11].

Усі вищеперераховані методи і засоби базуються на різноманітних фізичних принципах. Поплавкові спостереження часто допускають значні похибки вимірювань. Головним чинником цих похибок є вплив вітру, поправки на вплив вітру розроблені і вводяться в умовах помірного вітру до 5м/с. Також, слід врахувати те, що поплавки часто опереджають рух води, що напряму впливає на показник швидкості, який вимірюється поплавками. Гідрометричний млинок є найпопулярнішим девайсом для виміру швидкостей річкових потоків, в його основу входить принцип гідродинаміки[12].

Великомасштабне зображення часток (DLSPIN) дає змогу виміряти вектори швидкості в різних зонах, що утворюються в контрольованому водному просторі. Пари послідовних знімків допомагають оцінити поточну



швидкість в водному просторі. Наразі цей прилад не є ефективним для використання в природних умовах, як правило, він застосовується в лабораторних умовах, а саме для вивчення властивостей різних рідин[13].

Ультразвукове зондування вважається різновидом акустичного вимірювання. Його принцип дії полягає в посиленні імпульсів ультразвуку з 2 випромінювачів у напрямку течії і реверсивному напрямку течії і подальшою реєстрацією обох проміжків часу.

Зондування ультразвуком може виконуватись в обох напрямках, але на обмеженій глибині, також важливим є горизонтальне положення ультразвукового променя. Через ряд причин, а конкретніше значних математичних помарок, популярним цей метод не став.

Вимірювання швидкості течії тепловим способом потребує на інтенсивного обміну теплом між потоком і зануреним в нього приладом. Ці прилади здебільшого використовуються в лабораторіях. В справжніх природних умовах вони використовуються при природних дослідженнях. І знову ж, через значні математичні помарки, популярним цей метод не став.

Електромагнітні методи використовуються для визначення різниці потенціалів, які в свою чергу дозволяють виміряти час проходження води через електромагнітні контури. Електромагнітні методи вважаються перспективними для вимірювань швидкості в зарощених і деформованих руслах.

Вимірювання глибин може використовувати механічні, ручні та акустичні прилади. Ручними приладами є рейка, гідрометрична штанга, ручний лот.

### 1.3 Вибір і обґрунтування напрямку досліджень

Зважаючи на величезні обсяги інформації, глобальні системи моніторингу будуються за ієрархічною структурою де значна частина даних обробляється на локальному рівні.

Важливою функцією локальної системи моніторингу є прогнозування повеней.

Емпіричне моделювання може бути реалізовано двома шляхами:

- стохастичні методи
- регресійні моделі.

Стохастичні методи потребують значного обсягу апіорної інформації і допущення про стаціонарність випадкового процесу. Регресійні моделі будуються на методі найменших квадратів (МНК), і на методі групового урахування аргументів (МГУА) та на індуктивному методі самоорганізації моделей.

МНК-метод дає змогу визначити параметри регресійної моделі при відомій її структурі.

Використовуючи МГУА-метод та індуктивний метод самоорганізації моделей, можна визначити не тільки параметри регресійної моделі, але й її оптимальну структуру на множині даних, яка носить назву перевірної множини.

Характерною особливістю МГУА-методу є те, що для регресійних моделей високої розмірності досить складно виразити вихід моделі у явній формі по відношенню до її входів (незалежних змінних).

Реалізація алгоритмів індуктивного методу самоорганізації моделей для об'єктів великої розмірності та, коли степінь полінома регресійної моделі є високою, нашоувхується на проблему «прокляття розмірності».

Таким чином, напрямком дослідження є синтез структури комп'ютерної системи моніторингу і прогнозування рівня води рік.

#### 1.4 Висновки до розділу

1. Проведено аналіз існуючих методів і засобів моніторингу і прогнозування рівня води рік, технічну реалізацію системи здійснюють за допомогою системи пристроїв та відповідного програмного забезпечення,

що дозволяє збирати та обробляти інформацію про погодні умови та рівень води в обраній місцевості.

2. Розглянуто сучасні технічні засоби вимірювання метрологічних параметрів.

3. Обрано і обґрунтовано напрям досліджень, що дозволить прогнозувати зміну рівня води в залежності від отриманих погодних параметрів.

## 2 РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ НА ОСНОВІ ГЕНЕТИЧНИХ АЛГОРИТМІВ

2.1 Алгоритм опису процесів з некрatними частотами на основі генетичних алгоритмів

Дані, отримані при вивченні цілої низки екологічних процесів, як правило, подані у вигляді тимчасового ряду. Крім того, тимчасові ряди, які пов'язані з процесами в навколишньому середовищі, часто мають помітні сезонні або періодичні компоненти. Ці компоненти зазвичай змінюються в часі і, в більшості випадків, описуються сезонними статистичними моделями.

Видно, що структура моделі вже вибрана, і завдання лише у процедурі визначення ідентичності параметрів цієї моделі, яка базується, в більшості випадків, на методі найменших квадратів. Очевидно, що точність відповідних моделей і емпіричних даних залежить від структури моделі. Наприклад, при виборі моделі як рівняння регресії при збільшенні членів рівняння регресії похибка, що була визначена в усіх експериментальних точках, монотонно убуває. Після того, як число членів регресії зрівняється з числом експериментальних точок, похибка стає рівною нулю. Якщо протягом певного часу через ряд точок проходить велика кількість кривих, то за допомогою методу найменших квадратів можна створити нескінченно велику кількість моделей для даного числового ряду.

Згідно з теоремою Геделя з використанням точок послідовностей, які були використані для визначення коефіцієнтів за методом найменших квадратів, практично неможливо знайти оптимальну уніфіковану модель. Тільки зовнішні доповнення дозволяють знайти єдину оптимальну модель для даного критерію складності [14].

Вибір структури моделі здійснюється на основі типу процесу кривої. Постає питання про вибір базових функцій для математичної моделі. Наприклад, для коливальних процесів з нульовим середнім значенням

рекомендується використовувати гармонічні моделі, або якщо коливальний процес має тенденцію до зростання або спаду, то можна створити модель, яка являє собою суму полінома тренда і гармонічний баланс[15]. Моделі коливальних процесів, в яких не існує періодичного повторення, подані у вигляді кінцевої кількості гармонік з некротними частотами.

Розглянемо екологічний процес, який може бути поданий у вигляді математичної моделі, що містить лінійну тенденцію  $h(t)$ , гармонічну складову  $G(t)$  і залишкову складову  $H_t$ , а саме:

$$\tilde{H}_t = H_t + G(t) + h(t), \quad (2.1)$$

Таким чином, метод синтезу математичних моделей був розроблений на ідеях генетичних алгоритмів для моделювання складних коливальних процесів з некротними частотами. Розглянемо його застосування для екологічних процесів.

Гармонічна складова з некротними частотами, що притаманна досліджуваним процесам, також процесам з екологічним характером, моделюється таким виразом

$$G(t) = A_0 + \sum_{j=1}^m (A_j \sin(t\omega_j) + B_j \cos(t\omega_j)), \quad (2.2)$$

де  $t$  – такти відліку часу,  $t = 1, 2, 3, \dots, N$ ;

$A_0, A_j, B_j$  – параметри гармонічних рядів (2.2);

$\omega_j = \omega_{j-1} + \Delta\omega_j$  – некротні частоти,  $j = 1, 2, 3, \dots$

За даними спостережень процесу можна було б оцінити параметри ряду (2.2) необхідної реалізація стану  $N \geq 3m + 1$ , де  $N$  – такти відліку часу.

Сума декількох гармонік ряду (2.2), в яких коефіцієнти  $A_j, B_j$  визначаються за допомогою методу найменших квадратів, а число гармонік

і їх частоти обрані так, щоб отримати мінімум деякого зовнішнього критерію вибору, що називається гармонічною тенденцією оптимальної складності[16].

Визначення параметрів моделі (2.2) відбувається в кілька етапів.

Перша фаза є розрахунком коефіцієнтів балансу  $\alpha_p$ .

Нехай деяка функція  $g(t)$ , що є сумою  $m$  гармонічних складових з некротними частотами, визначається в інтервалі  $1 \leq t \leq N$ . Виберемо деяку фіксовану точку  $t=i$  і довільне  $p$ , запишемо:

$$g(i+p) = \sum_{j=1}^m (A_j \sin((i+p)\omega_j) + B_j \cos((i+p)\omega_j)), \quad (2.3)$$

$$g(i-p) = \sum_{j=1}^m (A_j \sin((i-p)\omega_j) + B_j \cos((i-p)\omega_j)). \quad (2.4)$$

Визначимо суму функцій (2.3) і (2.4), використовуючи відоме тригонометричне співвідношення

$$g(i+p) + g(i-p) = 2 \sum_{j=1}^m (A_j \sin(i\omega_j) + B_j \cos(i\omega_j)) \cos(p\omega_j). \quad (2.5)$$

Візьмемо суму всіх  $p=0, m-1$  лівої і правої частин рівняння (2.5) з ваговими коефіцієнтами  $\alpha_p$

$$\sum_{p=0}^{m-1} \alpha_p (g(i+p) + g(i-p)) = \sum_{p=0}^{m-1} \alpha_p 2 \sum_{j=1}^m (A_j \sin(i\omega_j) + B_j \cos(i\omega_j)) \cos(p\omega_j). \quad (2.6)$$

Порядок визначення потреби суми має бути змінений в правій частині останньої рівності

$$\sum_{p=0}^{m-1} \alpha_p (g(i+p) + g(i-p)) = \sum_{j=1}^m (A_j \sin(i\omega_j) + B_j \cos(i\omega_j)) \left( 2\alpha_0 + \sum_{p=0}^{m-1} 2\alpha_p \cos(p\omega_j) \right), \quad (2.7)$$

вагові коефіцієнти  $\alpha_p$  потрібно вибрати за зробленою умовою

$$\alpha_0 + \sum_{p=1}^{m+1} \alpha_p \cos(p\omega_j) = \cos(m\omega_j), \quad j = \overline{1, m}, \quad (2.8)$$

визначаємо

$$g(i+m) + g(i-m) = 2 \sum_{j=1}^m (A_j \sin(i\omega_j) + B_j \cos(i\omega_j)) \cos(m\omega_j). \quad (2.9)$$

Порівнюючи рівняння (2.7) (2.9) і з огляду на рівняння (2.26), отримаємо

$$\sum_{p=0}^{m-1} \alpha_p (g(i+p) + g(i-p)) = \sum_{j=1}^m (A_j \sin(i\omega_j) + B_j \cos(i\omega_j)) 2 \cos(m\omega_j) = g(i+m) + g(i-m). \quad (2.10)$$

Величина

$$b_i = g(i+m) - \sum_{p=0}^{m-1} \alpha_p (g(i+p) + g(i-p)) + g(i-m), \quad i = \overline{m+1, N-m} \quad (2.11)$$

характеризує точність, з якою коливальний процес виражається через задану суму гармонічних складових. Іншими словами, значення функції  $g(t)$  в певний момент часу, розташовані симетрично відносно довільної точки  $i$ , мають задовольняти отримане рівняння балансу. Якщо процес коливальний точно відображено в зазначеній кількості гармонічних складових, тобто,  $b_i = 0$ , рівняння (2.8) для довільної частоти запишеться так

$$\alpha_0 + \sum_{p=1}^{m-1} \alpha_p \cos(p\omega) = \cos(m\omega). \quad (2.12)$$

Використовуючи рекурентне співвідношення

$$\cos(p\omega) = 2 \cos((p-1)\omega) \cos \omega - \cos((p-2)\omega), \quad p = \overline{2, m} \quad (2.13)$$

зводиться до алгебраїчних рівнянь  $m$ -го степеня споріднення  $\cos \omega$

$$P_m z^m + P_{m-1} z^{m-1} + \dots + P_1 z + P_0 = 0, \quad (2.14)$$

де  $z = \cos \omega$ , коефіцієнти  $P_i$ ,  $i=0,1,\dots,m-1$  є функціями параметрів  $\alpha_p$ ,  $p=0,1,\dots,m-1$ ,

Рівняння (2.14) має  $m$  коренів, які однозначно визначають  $\omega_j$ ,  $j=\overline{1,m}$ .

Таким чином, щоб задатися параметрами  $A_0$ ,  $A_j$ ,  $B_j$  і  $\omega_j$ ,  $j=\overline{1,m}$  гармонічної тенденції, то спочатку мають бути визначені вагові коефіцієнти  $\alpha_p$ . Коефіцієнти балансу знаходять з умови мінімізації залишкових рівнянь балансу ординат

$$B = \sum_{i=m+1}^{N-m} b_i^2, \quad (2.15)$$

де  $b_i$  визначається рівнянням (2.15), в якому значення  $g$  відповідних дискретних аргументів замінюються на експериментальні дані.

Отже, завдання вирішується WII

$$\min_{\bar{\alpha}} J(\bar{\alpha}) = \sum_{i=m+1}^{N-m} \left( z_{i,m} - \sum_{p=0}^{m-1} \alpha_p g_{i,p} \right)^2, \quad (2.16)$$

де  $\bar{\alpha} = (\alpha_0, \alpha_1, \dots, \alpha_{m-1})^T$  – вектор вагових коефіцієнтів;

$$z_{i,m} = \bar{g}(i+m) + \bar{g}(i-m);$$

$$g_{i,p} = \bar{g}(i+p) + \bar{g}(i-p);$$

$T$  є символом транспонування матриць.

Завдання (2.16) буде представлено в матричній формі вектора



$$\min_{\bar{\alpha}} : J(\bar{\alpha}) = (\bar{z}_m - F_m \bar{\alpha})^T (\bar{z}_m - F_m \bar{\alpha}) , \quad (2.17)$$

$$\bar{z}_m = \begin{bmatrix} \bar{g}(2m+1) + \bar{g}(1) \\ \bar{g}(2m+2) + \bar{g}(2) \\ \dots \\ \bar{g}(N-m+1) + \bar{g}(N-m) \end{bmatrix} ;$$

де

$$F_m = \begin{bmatrix} 2\bar{g}(m+1) & \bar{g}(m+2) + \bar{g}(m) & \bar{g}(m+3) + \bar{g}(m-1) & \dots & \bar{g}(2m) + \bar{g}(2) \\ 2\bar{g}(m+2) & \bar{g}(m+3) + \bar{g}(m+1) & \bar{g}(m+4) + \bar{g}(m) & \dots & \bar{g}(2m+1) + \bar{g}(3) \\ \dots & \dots & \dots & \dots & \dots \\ 2\bar{g}(N-m) & \bar{g}(N-m+1) + \bar{g}(N-m-1) & \bar{g}(N-m+2) + \bar{g}(N-m-2) & \dots & \bar{g}(N-1) + \bar{g}(N-2m+1) \end{bmatrix}$$

.

Мінімізація (2.17) приводить до нормальних гаусових рівнянь в матричній формі, це виглядатиме так:

$$F_m^T F_m \bar{\alpha} = F_m^T \bar{z}_m , \quad (2.18)$$

З останнього рівняння можна знайти

$$\bar{\alpha} = (F_m^T F_m)^{-1} F_m^T \bar{z}_m , \quad (2.19)$$

Використовуючи формулу (2.18) можна тільки тоді, коли розмірність вектора  $\bar{\alpha}$  мала. Рівняння (2.18) має бути розв'язано одним з численних методів, наприклад, методом Гаусса зі зворотним потоком, якщо умова не задовольняється [17].

Рівняння (2.14) може бути розв'язано, якщо відомі вагові коефіцієнти

$\alpha_p$ , а розв'язок  $z$  забезпечує унікальну ідентифікацію частоти гармонік  $\omega_j$ ,  $j = \overline{1, m}$ .

Тепер завдання полягає в оптимальному синтезі гармонічного ряду (2.2).

Є два підходи, щоб вирішити цю проблему. Перший передбачає видалення гармонік в різних комбінаціях з повної серії  $\omega_j$ ,  $j = \overline{1, m}$ . Другий метод заснований на ідеї багаторядного алгоритму групи значних аргументів (MAGCA). Відповідно до цього способу кількість гармонік, які входять в модель, зростає до тих пір, поки це дозволяє знизити критерій вибору [18]. Найпростішим є алгоритм з послідовним випуском кращих моделей в кожній серії, але більш ефективним є алгоритм, коли кілька гармонік в кожному ряду виділяються.

Нехай отримано виконання вихідного значення процесу. Деяка частина цих даних, які містять  $N_R$ , це оглядові послідовні точки виділеної навчальної послідовності. Інші діляться на дві частини:  $N_Q$  – випробування і  $N_S$  – експертиза, всього точок  $N = N_R + N_Q + N_S$ .

Недоліком першого підходу до вирішення цієї проблеми є необхідність сортування для великого числа варіантів, що визначається як

$$\text{сума } S_v = \sum_{i=1}^m C_m^i = 2^m - 1,$$

$$\text{Ми знаємо, що } (a+b)^n = \sum_{i=0}^n C_n^i a^{n-i} b^i = a^n + \sum_{i=1}^n C_n^i a^{n-i} b^i, \text{ якщо } a=b=1$$

$$\text{, то } \sum_{i=1}^m C_m^i = 2^m - 1.$$

$$\text{Так, } S_v = 2^m - 1,$$

Наприклад, коли  $m = 30$ , потрібно сортувати 1073741823 варіанти, що потребує багато комп'ютерного часу. У другому підході, в результаті реалізації алгоритму мульти-MAGCA, неможливо отримати математичну модель в явному вигляді, і це є суттєвим недоліком цього способу.

Таким чином, для усунення проблеми великого масштабу і отримання форми моделі, було запропоновано використовувати інший підхід до

побудови математичних моделей коливальних процесів, яка заснована на ідеях генетичних алгоритмів.

Всі реалізації вихідного значення процесу або явища розділені на три частини в таких пропорціях:  $N_R=0,7N$ ,  $N_Q=0,2N$  і  $N_S=0,1N$ . Для даних

$N_R+N_Q$  вагові коефіцієнти  $\alpha_p$  визначені як розв'язки лінійних алгебраїчних рівнянь (2.18) за допомогою процесу усунення Гаусса з вибором головного елемента. Розв'язання рівняння (2.14) відносно змінної  $z$  дозволяє знайти частоту  $\omega_j$ ,  $j=\overline{1,m}$ , За відомими частотами  $\omega_j$  на множині точок  $N_R+N_Q$  потрібно знайти параметри моделі (2.2)  $A_0$ ,  $A_j$  і  $B_j$ .

Завдання буде вирішено за допомогою генетичних алгоритмів. Форма впорядкованої структури з довжиною  $m$ , де  $i$ -те місце буде змінено на нуль або один, це залежить від того, як частота  $\omega_j$  видаляється з обраної повної серії  $m$  вправо або вліво. В теорії генетичних алгоритмів така впорядкована послідовність відома як хромосоми або окремі особи, а також атомний елемент хромосом (один або нуль) – ген. Набір хромосом утворює популяцію. Важливою в теорії генетичних алгоритмів є функція пристосування, що визначає ступінь адаптації особин в популяції[19]. Це дозволяє вибрати з усієї сукупності осіб, які є найбільш пристосованими, тобто такими, які мають найбільше (найменше) значення функції регулювання.

У задачі модельних процесів синтезу осцилювальних, функція адаптації є об'єднанням критеріїв відбору

$$\rho = \sqrt{n_d^2 + B^2}, \quad (2.20)$$

де  $n_d^2$  – критерій зсуву, який розраховується за формулою

$$n_d^2 = \frac{\sum_{i=1}^N (g_i(R) - g_i(S))^2}{\sum_{i=1}^N \bar{g}_i^2}, \quad (2,21)$$

де  $B$  – залишкова функція, яка визначається як (2.15);

$g_i(R)$ ,  $g_i(S)$  – величини, значення яких обчислюються відповідно до кількості точок  $N$  за формулою (2.2), і коефіцієнти моделі (2.2) знайдені згідно з номерами  $N_R + N_Q$  и  $N_S$ .

Таким чином, завдання синтезу моделей коливального процесу буде формуватися таким чином: вибрати хромосому вихідної популяції хромосом шляхом еволюційного відбору, що забезпечує оптимальне значення функції адаптації (мінімальне значення критерію вибору (2.21)).

## 2.2Складові генетичного алгоритму

### S1, Формування вихідної популяції (ініціалізація)

На першому кроці алгоритму випадковим чином формується населення  $I$  індивіди, кожен з яких має хромосоми з довжиною  $m$ .

### S2, Оцінювання адаптивності хромосоми в популяції

Для кожного критерію відбору хромосом обчислюються (2.21). Ця процедура виконується в нижчеописаний спосіб. Відповідно до моделі (2.2) утворена матриця

$$F = \begin{bmatrix} 1 & \sin \omega_1 & \cos \omega_1 & \sin \omega_2 & \cos \omega_2 & \dots & \sin \omega_m & \cos \omega_m \\ 1 & \sin(2\omega_1) & \cos(2\omega_1) & \sin(2\omega_2) & \cos(2\omega_2) & \dots & \sin(2\omega_m) & \cos(2\omega_m) \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & \sin(N\omega_1) & \cos(N\omega_1) & \sin(N\omega_2) & \cos(N\omega_2) & \dots & \sin(N\omega_m) & \cos(N\omega_m) \end{bmatrix}.$$

Це необхідно для подвійних одиниць і нулів в поточній хромосомі. Наприклад, якщо генерується хромосома на першому етапі:  $Ch=[1001011]$ , Після подвоєння операції, вона буде мати вигляд

$$Ch_d = \left[ 11 \ 00 \ 00 \ 11 \ 00 \ 11 \ 11 \right].$$

Оскільки модель (2.20) завжди має коефіцієнт  $A_0$ , то має бути доданий один ген в хромосомі  $Ch_d$  на першій позиції. В результаті буде отримано  $Ch_{d0}=[111000011001111]$ .

Необхідність подвійної операції пояснюється тим, що пара коефіцієнтів  $A_j$ ,  $B_j$  несе відповідальність кожної частоти  $\omega_j$ . Відповідно до існуючої хромосомі  $Ch_{d0}$  з матриці  $F$  створити нову матрицю  $F_{new}$ , видаливши стовпці з матриці  $F$ , що пов'язані з хромосомами нулями  $Ch_{d0}$ , дві матриці  $F_R$  і  $F_S$  буде створено з отриманої матриці розмірів  $(N_R+N_Q) \times m_1$  і  $N_S \times m_1$  [20]. Матрицю  $F_R$  створюють перші  $N_R+N_Q$  стовпці матриці  $F_{new}$ , а матрицю  $F_S$  – останні  $N_S$  стовпці матриці  $F_{new}$ .

Про числі точок  $N_R+N_Q$  і  $N_S$  підраховують коефіцієнти, відмінні від нуля  $A_0$ ,  $A_j$  і  $B_j$  моделі (2.20), розв'язуючи нормальне рівняння Гаусса

$$F_R^T F_R \bar{A}_R = F_R^T \bar{g}_R, \quad (2,22)$$

$$F_S^T F_S \bar{A}_S = F_S^T \bar{g}_S, \quad (2,23)$$

де  $\bar{A}_R = (A_0^{(R)}, A_1^{(R)}, B_1^{(R)}, A_2^{(R)}, B_2^{(R)}, \dots, A_{m_1}^{(R)}, B_{m_1}^{(R)})$ ,

$\bar{A}_S = (A_0^{(S)}, A_1^{(S)}, B_1^{(S)}, A_2^{(S)}, B_2^{(S)}, \dots, A_{m_1}^{(S)}, B_{m_1}^{(S)})$  – вектори параметрів моделей, які пов'язані з іншою хромосомою від вихідної популяції і розраховані відповідно за формулами (2.22) і (2.23);

$\bar{g}_R = (\bar{g}^{(1)}, \bar{g}^{(2)}, \dots, \bar{g}^{(N_R+N_Q)})^T$ ,  $\bar{g}_S = (\bar{g}^{(1)}, \bar{g}^{(2)}, \dots, \bar{g}^{(N_S)})^T$  – вектори

експериментальних даних про кількість точок  $N_R+N_Q$  і  $N_S$ .

При певному наборі коефіцієнтів  $\bar{A}_R$  и  $\bar{A}_S$  моделі (2.2) за числом точок  $N$  підраховують  $g^{(R)} = F_{new} \bar{A}_R$ ,  $g^{(S)} = F_{new} \bar{A}_S$ , (2.24)

За допомогою формули (2.20) обчислюються критерії вибору відповідно до (2.11) і (2.15). Критерій вибору значення обчислюється для кожної хромосоми і результат являє собою набір значень  $\rho_i$ ,  $i = \overline{1, M}$ , де  $M$  – число хромосом в популяції.

### S3, Перевірка стану зупинки алгоритму

Визначити  $\rho_{\min} = \min_{i \in M} \rho_i$ , (2.25)

Якщо мінімальне значення (2.25) і критерій вибору (2,20) не перевищує деяке позитивне значення  $\varepsilon$ , виконання алгоритму зупиняється. Зупинка алгоритму також може статися, якщо виконання не приводить до поліпшення функції або налаштування, коли алгоритм завершив заданий число ітерацій.

Після операції подвоєння та приєднання одного гена на хромосомі  $Ch_d^i$  у нього є  $Ch_{d0}^i$ , Ця хромосома визначає структуру моделі оптимальної складності і так створює матрицю  $F^*$ : стовпці, які пов'язані з нульовими генами хромосом  $Ch_{d0}^*$ , видаляються з вихідної матриці  $F$ .

Перетворення параметрів моделі (2.2) відбувається на множині всіх точок вихідного набору даних[21].

*S4*, Вибір хромосом.

За значеннями функції адаптації, які розраховуються на другій стадії, відбувається відбір хромосом, які будуть брати участь у створенні нащадків для наступної популяції. Цей вибір зроблено відповідно до принципу природного відбору як кращий шанс створити нову популяцію хромосом з кращими функціями регулювання значення, тобто такими, які забезпечують мінімальне значення критеріїв відбору (2.2).

Найбільш поширені методи відбору являють собою метод рулетки і турнір методу відбору.

Метод рулетки Sures полягає в тому, що особи вибираються для нового покоління пропорційно їх значенням функції адаптації[22]. Кожна хромосома отримує в околі своєї сім'ї багато копій, що задаються

$$E(Ch_i) = P(Ch_i) \cdot I, \quad (2.26)$$

де  $P(Ch_i)$  – ймовірність вибору хромосом;

$I$  – число хромосом  $Ch_i$ ,  $i = \overline{1, I}$  в популяції.

Значення  $P(Ch_i)$  розраховується за формулою

$$P(Ch_i) = \frac{\rho(Ch_i)}{\sum_{i=1}^I \rho(Ch_i)}, \quad (2.27)$$

де  $\rho(Ch_i)$  – значення функції адаптації  $Ch_i$  хромосом (в контексті завдання, коли вважається, що критерій вибору (2.2)).

Очевидно, що метод рулетки може бути використаний, коли функція адаптованості позитивна і може бути використана тільки в задачах максимізації.

Турнірний метод може бути використаний в задачах максимізації, а також в завданні звести до мінімуму функції. У турнірі відбору всі хромосоми діляться на підгрупи, а потім вибираються з кожної підгрупи утворені хромосоми з кращого технологічності. Підгрупи можуть мати будь-який розмір, але більша частина населення ділиться на підгрупи з 2–3 особин в кожній з них.

#### S5, Формування нової популяції нащадків

Формування популяції нащадків здійснюється за допомогою двох основних операторів – перетину і мутації. Слід зазначити, що оператор мутації відіграє другорядну роль порівняно з оператором переходу. Це означає, що в генетичному алгоритмі перетин виконується майже завжди, а мутація – рідко. Імовірність перетину є досить великою ( $0,5 \leq P_c \leq 1$ ) в той час як ймовірність мутацій обрана досить малою ( $0 \leq P_m \leq 0,1$ ).

Оператор ймовірності мутації змінює значення гена в хромосомі, протилежній одній, а саме: від 1 до 0, або від 0 до 1. Імовірність мутації  $P_m$  може бути випадковим вибором чисел з інтервалу  $[0; 1]$  для кожного гена і відбір генів для цієї активності, число яких буде меншим або дорівнює  $P_m$ . Мутації можуть бути виконані на басейн родичів і басейн нащадків.

Оператор переходу складається з двох етапів. У першій фазі підгрупи  $z$  формуються окремі особи, де вибирається найкраща хромосома критерію вибору  $R(Ch_{do}^i) = \min_i R(Ch_{do,i})$ , Результатом є нова популяція хромосом, які оператор використовує на другому етапі.

Перетин виконується на другому етапі. З басейну родичів  $M(k)$  випадковим чином з ймовірністю  $P_c$  утворюються пари. Пара хромосом вибирається випадковим чином з популяції. Номер  $P_c$  генерується випадковим чином з інтервалом  $[0; 1]$ , і якщо значення не більше  $P_c$ , пара хромосом проводить спарювання. В іншому випадку пара хромосом не



змінюється. Потім для кожної пари родичів позиції гена (локус) у хромосомі визначають точку перетину. Якщо кожна хромосома сім'ямістить в собі  $m$  генів, перетин точки  $L_c$  – позитивне ціле число менше  $m$ . Таким чином, фіксуючи точку перетину зводять до випадкового вибір цілого числа з інтервалу  $[1; L_c - 1]$ .

Дія оператора схрещування приводить до того, що пара родичів утворюють нову пару потомства так: перший нащадок пари хромосом в позиціях від 1 до  $L_c$  складається з генів першого відносного, а на позиціях від  $L_c + 1$  до  $m$  – з генів другого родича; другий нащадок пари хромосом на позиціях від 1 до  $L_c$  складається з генів другого відносних, і на позиції від  $L_c + 1$  до  $m$  – з генів першого родича.

Після перетину оператор переходить до S2.

Отже, таким чином буде отримана математична модель оптимальної складності коливального процесу з некратними частотами і з використанням ідеями генетичних алгоритмів.

### 2.3 Висновки до розділу

1. Було розроблено математичну модель на основі генетичних алгоритмів, що дозволить оптимізувати та покращити процес збору та обробки інформації про погодні умови та рівень води.

2. Проаналізовано синтез моделей коливального процесу та його формування, враховано вагові коефіцієнти і на їх основі обрано хромосоми вихідної популяції шляхом еволюційного відбору, що забезпечує оптимальне значення функції адаптації.

### **3 РОЗРОБКА ПРОГРАМНОГО МЕТОДУ ОПТИМІЗАЦІЇ КОМП'ЮТЕРНОЇ СИСТЕМИ ПРОГНОЗУВАННЯ РІВНЯ ВОДИ РІЧОК**

3.1 Синтез структури комп'ютерної системи моніторингу і прогнозування рівня води в річці

Метод прогнозування на основі генетичних алгоритмів був запропонований і на його основі було реалізовано комп'ютерну систему моніторингу та прогнозування рівня води.

Автоматизована комп'ютерна система моніторингу і прогнозування рівня води в річці була розроблена з метою ефективної експлуатації та зниження втрат, спричинених негативними наслідками від повеней, рішення приймалися з урахуванням отриманої інформації.

Комп'ютерна система моніторингу і прогнозування призначена для автоматичного збору інформації про погоду в районі дослідження рівня води в річці та її передачі її від точок спостереження до контролера робочої станції, де формується відповідна база даних метеодосліджень, а вже на основі цих даних, модуль спеціального програмного забезпечення виконує прогнозування водного режиму річки[23].

Завдання, які вирішуються за допомогою комп'ютерної системи:

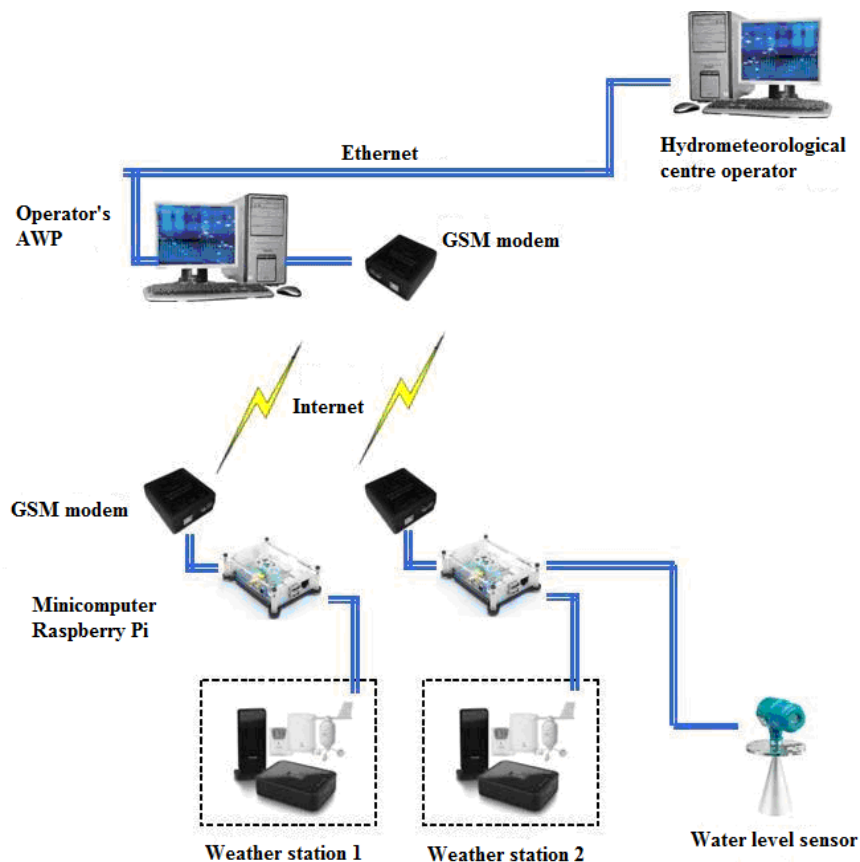
- дослідження впливу погодних умов на рівень води в річках за допомогою дистанційного моніторингу;
- аналіз і оцінювання річкового стоку води;
- прогнозування стану річки(за спостереженнями);
- управління повенями.

Розроблена комп'ютерна система виконує функцію моніторингу і прогнозування, тому вона інтегрує підсистеми збору і обробки інформації про погодні умовита рівень води в річках і підсистем аналізу та прогнозування рівнів води на основі даних. Підсистеми під'єднані до однієї

комп'ютерної системи за допомогою каналів зв'язку для забезпечення швидкого обміну даних між ними.

Ієрархія системи моніторингу та прогнозування рівнів води в річках складається з чотирьох рівнів (рис. 3.1). Перший і другий рівні системи містять в собі вимір і збір даних для моніторингу погодних умов і рівня води. Треті і четверті рівні об'єднані в підсистему аналізу і прогнозування рівня води в річці.

Низький рівень системи забезпечує набір датчиків метеорологічних станцій і датчик рівня, який встановлюється безпосередньо в обраних точках спостереження на досліджуваній ділянці[24]. Цей рівень системи забезпечує збір і передачу на більш високий рівень інформації про величину вимірюваних параметрів погоди: температуру повітря, вологість, швидкість і напрямок вітру, атмосферний тиск і опади[25]. Крім збирання цих параметрів система виконує вимірювання рівня води на річці за допомогою рівнемірів РЛС.



### Рисунок 3.1 – Структурна схема комп'ютерної системи

На другому рівні є окремі міні-ЕОМ (контролери), які виконують локальне управління обладнанням в точках спостереження. Це забезпечує перевірку режимів роботи датчика і повідомлення в разі їх відмови. Устаткування такого рівня також отримує інформацію про параметри вимірюваних величин погодних умов і рівень води в річках, обробляє цю інформацію і відправляє його на наступний рівень[26].

Третій рівень складається з персонального комп'ютера і спеціального програмного забезпечення, також має оптимізоване робоче місце оператора (АРМ). АРМ оператора призначене для обробки, аналізу і збереження в базі даних прийнятих відомостей про погодні умови та поточний стан річки[27]. На цьому рівні (на базі отриманих метеоданих) обчислюється прогнозований рівень води в річці. Використовуючи спеціальне програмне забезпечення АРМ оператора, що базується на основі методу побудови емпіричних моделей на генетичних алгоритмах, дозволяє контролеру виконувати віддалений моніторинг і прогнозування стану навколишнього середовища.

Деяка інформація з АРМ оператора у вигляді звітної документації надається операторам Гідрометеоцентру для аналізу і прийняття рішень.

#### 3.2 Обґрунтування і вибір операційної системи програмного забезпечення

Комп'ютерна система підтримки прийняття рішень була створена для реалізації розробленого методу. Основним елементом цієї комп'ютерної системи є оптимізоване робоче місце оператора (АРМ) Basen на серійному ПК, який підключений до локальної мережі Ethernet для отримання даних про погоду (рис.3.2). Основа АРМ оператора – це спеціальне програмне забезпечення, яке, відповідно до розроблених алгоритмів, дозволяє збирати дані про водний режим річок і погодні умови, оперативне отримання,

зберігання і обробка зібраної інформації для встановлення поточного стану річки дозволяє отримати достовірні прогнози з подальшим відображенням на екрані дисплея результатів такої обробки[28].

Для ефективної роботи комп'ютерної системи спеціальне програмне забезпечення дозволяє виконувати такі завдання:

а) приймати дані від датчиків рівня води і метеорологічних станцій автоматично, код чергового (інтерв'юера) не має залежати від типу датчиків (для забезпечення абстракції);

б) працювати з реляційною базою даних (зберігання даних, що надходять від датчиків про погодні умови, рівні води та результати прогнозованого рівня води);

в) використовувати існуючий код Matlab, який заснований на методі синтезу математичних моделей прогнозування рівнів води залежно від погодних умов.

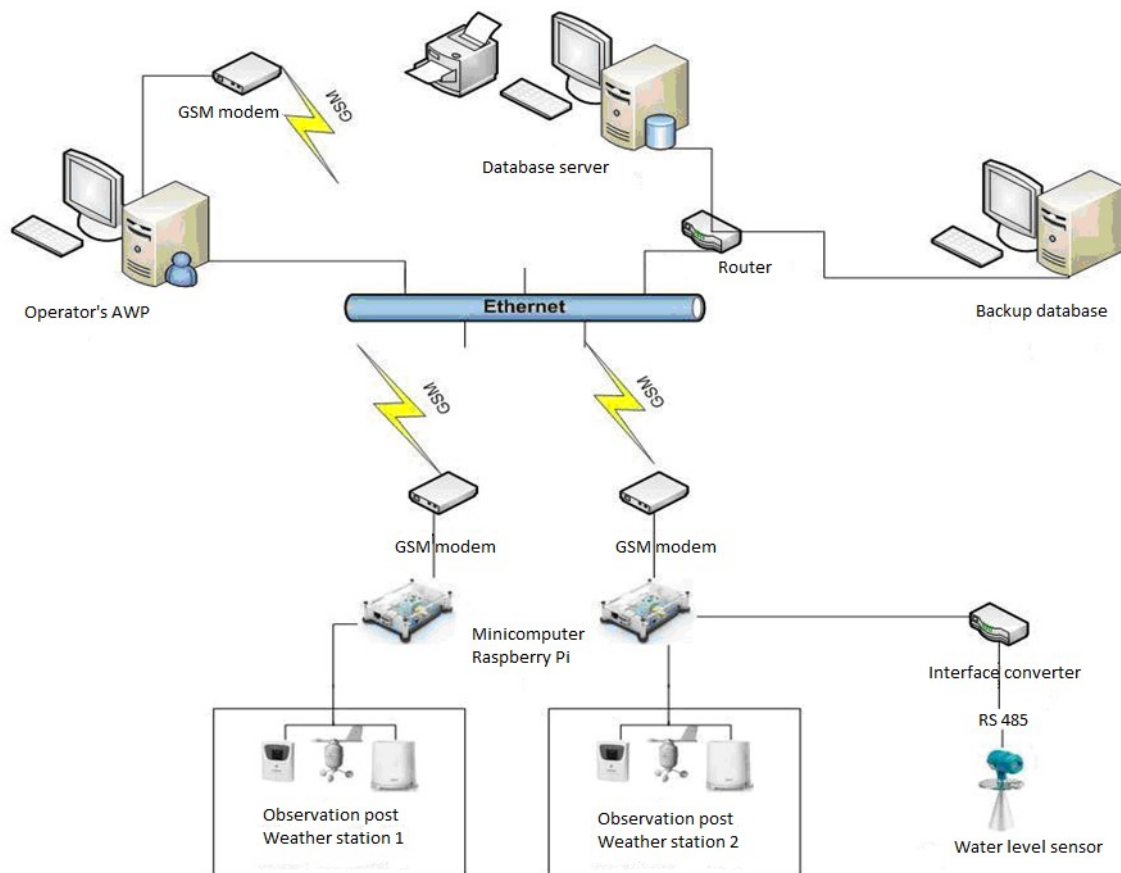


Рисунок 3.2 – Комп'ютерна система прогнозування Дністра

Був розроблений спеціальний програмний модуль (з урахуванням вищевказаних вимог до програмного забезпечення), який керує процесом прогнозування рівня води за зібраними даними з датчиків на погодні умови у річці Дністер. Програма наведена в Додатку В.

На рис. 3.3 показано взаємодію блока програм, які забезпечують поетапний процес прогнозування рівня води в річці Дністер.

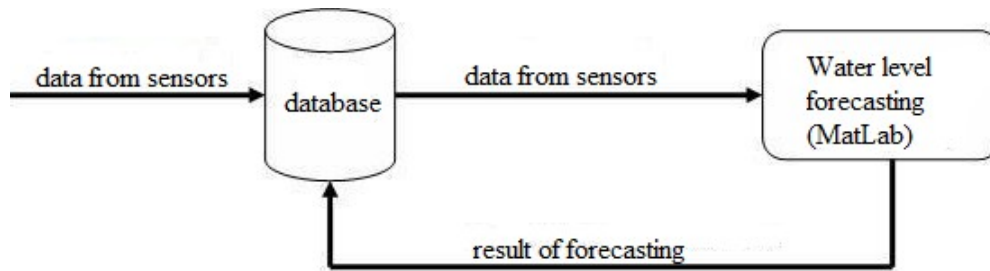


Рисунок 3.3 – Схема трафіку даних

Інформаційна база системи комп'ютерної підтримки прийняття рішень являє собою базу даних зібраної інформації про рівень води в річці Дністер і погодні умови. База даних генерується автоматично і зберігає поточні дані про параметри погодних умов, а саме: температуру, вологість, атмосферний тиск, опади, напрямок вітру і дані рівня води (див. рис.3.2).

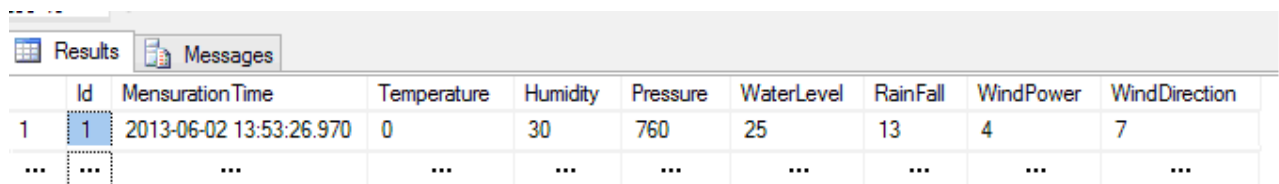
Актуальна інформація та архів бази даних вводяться в ОЗП для обробки і прогнозування за розробленим алгоритмом і відтворення у вигляді графіків, таблиць та текстової інформації.

Matlab-модуль проводить розрахунки прогнозованого рівня води в річці відповідно до розробленого математичного забезпечення. Результати, отримані за допомогою розрахунків, зберігаються в базі даних. Пізніше вони будуть розглянуті в плануванні заходів і рішень щодо запобігання стихійних лих, пов'язаних з повеннями.

Отже, комп'ютерна система отримання інтелектуального рішення заснована на результатах прогнозування рівня води в річці за допомогою програмного забезпечення й містить в собі наступні етапи (рис. 3.3.).

1. Опитування датчиків проводиться через певні періоди часу (час вказано в налаштуваннях програми (одна секунда)), потрібно запросити результати вимірювань датчиків.

2. Дані вимірювань від датчиків перетворюються в текстові дані об'єкта і зберігаються в базі даних як впорядковані записи з датою спостереження (рис. 3.4).



	Id	MensurationTime	Temperature	Humidity	Pressure	WaterLevel	RainFall	WindPower	WindDirection
1	1	2013-06-02 13:53:26.970	0	30	760	25	13	4	7
...	...	...	...	...	...	...	...	...	...

Рисунок 3.4 – Запис даних вимірювань в базу

3. Результати вимірювань зчитуються з бази даних і записуються в пам'ять комп'ютера при запуску основних додатків.

4. Програмний модуль запускає оболонку Matlab і виконує розрахунки відповідно до методу синтезу математичних моделей прогнозування рівнів річок.

5. Прогнозований результат рівня води реєструється в базі даних і передбачуваний стан річки оцінюється з використанням інтелектуального програмного забезпечення.

6. Відповідні ефективні операційні рішення створені відповідно до отриманого прогнозованого стану річки з виходом на дисплей монітора АРМ оператора.

Було розглянуто вимоги апаратних засобів і програмного забезпечення розробленої системи: від швидкості обробки при роботі з декількома

пристроями, доступними для різних платформ вибору мови програмування, до розробки програмного модуля даних.

Лідерами серед мов високого рівня для написання швидкого та ефективного програмного забезпечення є мови програмування: C ++, Java і C #.

C ++ дозволяє створювати дуже компактне і швидке рішення, але вимагає великих витрат на розробку, рішення для кожної платформи має бути записано окремо. Java є другою за швидкістю, але має перевагу в сумісності з різними платформами. C # є останньою мовою програмування, яка поєднує в собі міць C ++ і надійність Java. Мова програмування C # була створена Microsoft для підтримки нової платформи .Net., вона крос-платформена, як Java.

Мова програмування C # була обрана для написання коду системи, програмний код наведено Додатку В.

Основа програми буде складатися з певного числа об'єктів і правил взаємодії між ними, оскільки вибирається об'єктно-орієнтований підхід. Послуги охоплюватимуть набір як загальних, так і специфічних для кожного об'єкта властивостей, будуть взаємодіяти, посилаючи один одному повідомлення в широковідомому форматі.

Як відомо, існує вісім можливих напрямків вітру: північ, північний схід, схід, південний схід, південь, південний захід, захід, північний захід. У цьому випадку, щоб оголосити змінну «напрямок вітру», що містить список попередньо відомих значень, потрібно використовувати enum,

```
перерахування напрямку вітру
{
    Північ,
    Північний Схід,
    Схід,
    Південний Схід,
```



```

Південь,
Південний Захід,
Захід,
Північний Захід
}

```

Іменовані константи, які входять в список значень, мають базовий тип, який може бути будь-який іншим, ніж рядки. Для заданого змінного «напрямок вітру» базового типу є цілим числом типу (INT), За замовчуванням перше значення в списку Північ – це 0, друге Північний схід –1, і кожне наступне значення змінного напрямку вітру збільшується на 1.

Дані всіх датчиків записуються в базу даних. Тип і значення вимірювання меж результатів датчиків, які є вхідними параметрами математичної моделі, визначено, щоб створити структуру запису бази даних:

- Температура повітря (від -50, до +50 °С);
- Вологість повітря (0–100%);
- Атмосферний тиск (600–800 мм рт. ст.);
- Рівень води (-5 м до 5 м від базової лінії);
- Оподи;
- Напрямок вітру.

	Column Name	Data Type	Allow Nulls
PK	Id	int	<input type="checkbox"/>
	MensurationTime	datetime	<input type="checkbox"/>
	Temperature	int	<input type="checkbox"/>
	Humidity	int	<input type="checkbox"/>
	Pressure	int	<input type="checkbox"/>
	WaterLevel	int	<input type="checkbox"/>
	RainFall	int	<input type="checkbox"/>
	WindPower	int	<input type="checkbox"/>
	WindDirection	int	<input type="checkbox"/>
			<input type="checkbox"/>

### Рисунок 3.5 –Панель управління SQL

Перевірка даних здійснюється за допомогою атрибутів .NET.

Перший етап програмного модуля: працює інтерв'юер датчиків. Інтерв'юер надсилає запит про виміряні параметри (рис. 3.6). З певною частотою, яка може бути змінена в налаштуваннях програми, дані можуть бути від рівня води і метеорологічних датчиків.

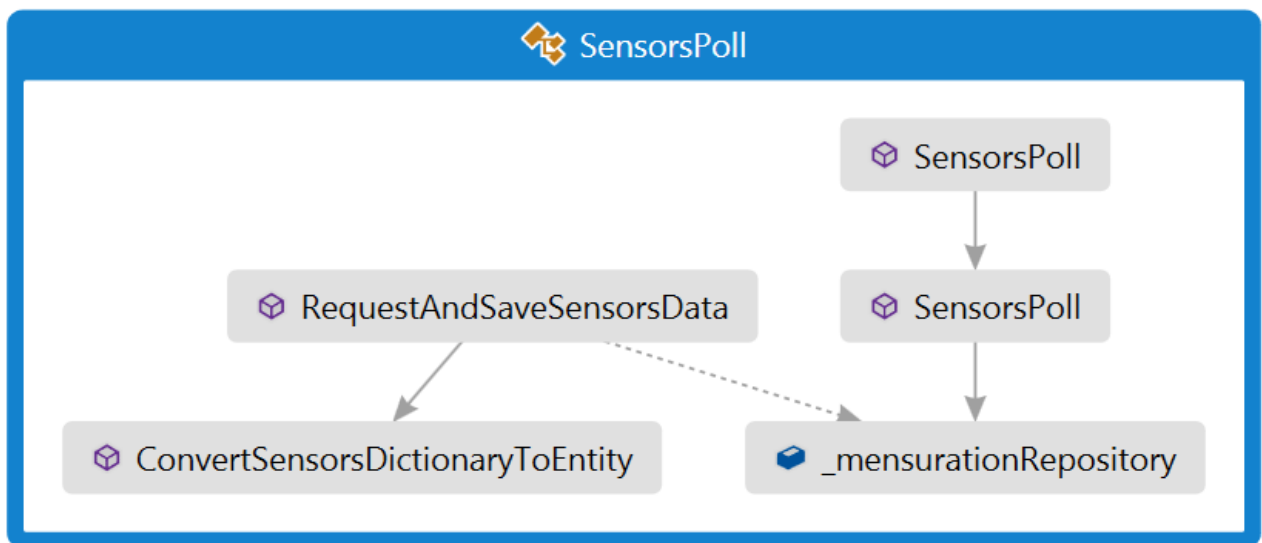


Рисунок 3.6 – Структура інтерв'юера датчиків

Отримані від інтерв'юера дані зберігаються в базі даних (метод RequestAndSaveSensorsData). Сховище бази даних підключено до конструктора, опитування датчиків шляхом ін'єкцій з тим, щоб мати можливість роботи з базою даних. Ін'єкція в конструкторі є підвидом ін'єкції залежностей.

Dependency Injection є забезпеченням зовнішніх компонентів залежно від програмного забезпечення. Це специфічна форма «інверсія управління», яка змінює зв'язок замовлення і являє собою спосіб отримання необхідної залежності.

Впровадження залежностей є більш гнучким, оскільки стає простішим створити альтернативну реалізацію цього виду послуг, а також вказати,

реалізацію яких потрібно використовувати, наприклад, у файлі конфігурації без змін в об'єктах, які використовують цю послугу.

Виконавчий блок програми зчитує інформацію з бази даних і перетворює її в файл. Програмний модуль Matlab починає працювати, якщо отримано достовірні дані про погоду для прогнозування рівня води. Matlab-модуль обчислює прогнозований рівень води за розробленим методом. Коли результати передбачення були отримані, програма перетворює інформацію з вихідного файлу і записує її в базу даних (рис. 3.7).

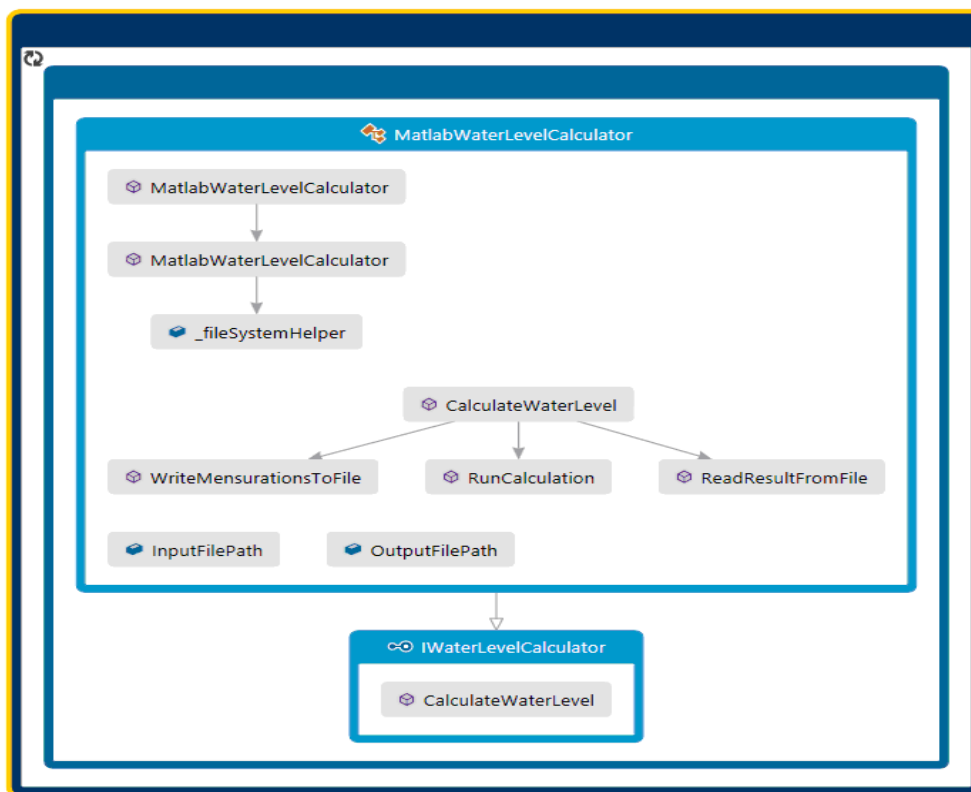


Рисунок 3.7 – Структура програми виконавчого блока

Основні функції програмного коду (Додаток В):

`PublicvoidRequestAndSaveSensorsData (objectstate)` – функція, яка запускається періодично, а також запитує дані від датчиків і зберігає їх в базі даних, не повертаючи ніяких значень;

`PublicstaticDictionary<string, Int> GetData ()` – функція, яка повертає поточні дані з датчиків як словник;

`Privatestatic MensurationEntity ConvertSensorsDictionaryToEntity (Dictionary <string, Int>data)`

– функція, яка перетворює дані від датчиків і повертає об'єкт вимірювання;

`string ReadFromFile (string Filepath)` – функція, яка зчитує інформацію з текстового файлу; Шлях до файлу передається як параметр, а вміст файлу повертається у вигляді рядка;

`voidWriteToFile(string Filepath, stringdata)` – функція, яка записує файл у текст; дані і шлях до файлу передаються як параметри;

`int CalculateWaterLevel (ICollection <MensurationEntity> mensurations)` – функція, яка повертає значення прогнозованого рівня води на основі вимірів;

`PublicvoidSaveMensuration (MensurationEntity mensurationEntity)` – функція, яка зберігає дані вимірювань про прогнозований рівень води в базі даних, не повертає ніяких значень.

### 3.3 Вибір технічних засобів моніторингу та прогнозування рівня води в річках

Комп'ютерна система апаратних засобів, які призначені для вимірювання характеристик навколишнього середовища в точках спостереження, складається з датчика рівнеміра KROHNE OPTIWAVE 5200 C/F і метеостанції OREGON LW301.

Датчики KROHNE OPTIWAVE 5200 C/F використовуються, в основному, для безперервного і безконтактного вимірювання рівня і об'єму рідин, паст, суспензій, деяких сипучих матеріалів в резервуарах-сховищах і промислових контейнерах, а також рівня потоку у відкритих каналах. Датчик KROHNE OPTIWAVE 5200 C/F з високою точністю ( $\pm 5$  мм) визначає відстань до поверхні досліджуваного середовища і має можливість запису віддаленого сигналу на пристроях (контролерах) автоматизованої

системи управління. Цей датчик рівня автоматизує управління і контроль процесу у виробництві[29].

У нашому випадку в комп'ютерній системі моніторингу і прогнозування рівня води в річці датчик KROHNE OPTIWAVE 5200 C/F використовується для вимірювання рівня води в річці Дністер.

Датчик KROHNE OPTIWAVE 5200 C/F (рис. 3.8) генерує і випромінює за допомогою антени радіосигнал, відбитий від поверхні води, що повертається через деякий час  $t$ . Таким чином, вимірювання відстаней до

поверхні визначається залежністю  $h = \frac{c \cdot t}{2}$ , де  $c$  – швидкість світла[30].

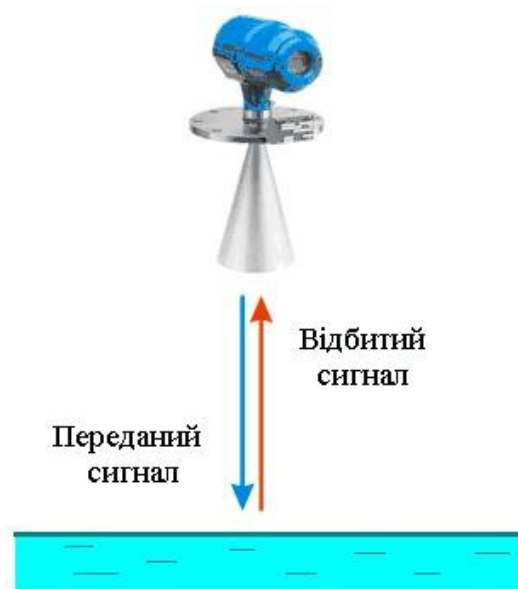


Рисунок 3.8 – Датчик рівня KROHNE OPTIWAVE 5200 C/F

Оскільки датчик KROHNE OPTIWAVE 5200 C/F дає можливість встановити тільки відстань до поверхні води, то, щоб визначити рівень води в річці, потрібно використовувати формулу

$$\tilde{H}_t = h_0 + h_o - h_m, \quad (3.1)$$

Де  $h_0$  – базовий рівень води в річці, м;

$h_o$  – значення датчика, які відповідають значенням  $h_0$ , м;

$h_m$  – значення струму датчика рівня, м.

Схематично визначення рівня річки показано на рис. 3.9.

По-перше, базовий рівень води в річці фіксується, щоб визначити поточний рівень води в річці  $h_o$  шляхом вимірювання доріжки при нормальних погодних умовах досліджуваного району. Датчик KROHNE OPTIWAVE 5200 C/F при тих же погодних умов визначає відстань до базової лінії –  $h_o$ . Таким чином, фіксуючи значення поточного рівня  $h_m$ , з використанням рівня датчика, рівень води в річці може бути знайдений за формулою (3.1).

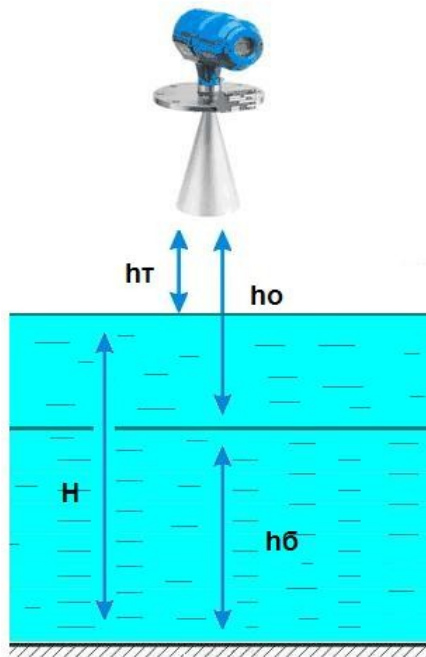


Рисунок 3.9 –  
Вимірювання рівня  
води в річці

Датчик  
KROHNE  
OPTIWAVE 5200 C  
/ F встановлено на  
автодорожньому  
мості через річку

Дністер (село Нижнів, Тлумацький район). кріплення датчика рівня має бути розташовано ближче до середини мосту, оскільки надійні дані щодо рівня води в річці можуть бути встановлені в цьому місці. Відповідно до технічних характеристик датчика KROHNE OPTIWAVE 5200 C/F діапазон виміру становить 30 м. За необхідності можуть бути використані

подовження антени. Схема датчика кріплення на мосту показана на рисунку 3.10.

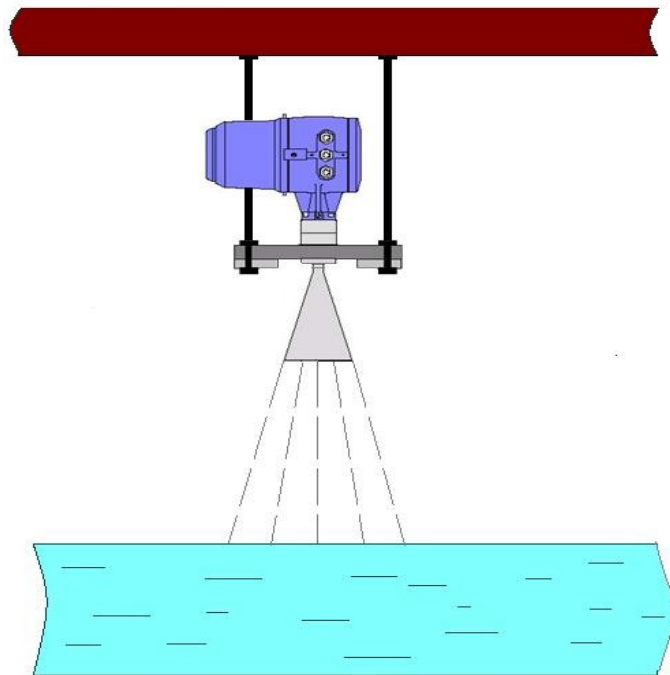


Рисунок 3.10 – Кріплення датчика рівня KROHNE OPTIWAVE 5200 C/F на мосту

Датчик KROHNE OPTIWAVE 5200 C/F підключений до міні-комп'ютера на основі плати «Малина» Pi для дистанційної передачі сигналу рівня води в річці (рис. 3.11). В комп'ютерній системі використовується wifi-інтерфейс RS-485, який сумісний з датчиком KROHNE OPTIWAVE 5200 C/F. Цей інтерфейс дозволяє зробити двосторонній зв'язок (півдуплексна передача) з системою. Комп'ютер посилає запит, і рівень датчика KROHNE OPTIWAVE 5200 C/F передає інформацію щодо рівня води у відповідь. Будь-який рівень датчика KROHNE OPTIWAVE 5200 C/F може бути налаштований з ПК (необхідний перетворювач RS-485/RS-232) з використанням програмного забезпечення PC-CAT. Максимальна довжина кабелю без підсилювача складає 2 км[30].

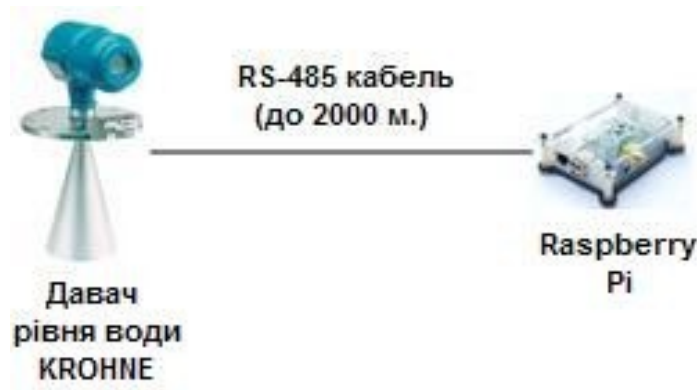


Рисунок 3.11 – Схема підключення датчика рівня KRONNE до плати «Малина» Pi комп'ютера

Метеорологічні датчики Орегон LW301 використовуються для визначення параметрів погоди в обраній області. Кілька точок спостереження з існуючих метеостанцій OREGON LW301 розташовані вгору по річці Дністер[30].

Бездротова погодна станція Oregon LW301 за допомогою датчиків вимірює температуру повітря, вологість, швидкість і напрям вітру, дані атмосферного тиску і опадів, і передають ці сигнали на приймач WH200 (рис. 3.12). Допускається підключати до 8 датчиків температури і вологості. Допустима відстань бездротових датчиків від приймача становить 100 м. Підключення проводиться у внутрішньому інтерфейсі метеостанції OREGON LW301.

Метеостанція OREGON LW301 має можливість передачі даних про погоду в Інтернет. Виміряні дані від бездротових датчиків зчитуються сигналами приймача WH200 і передаються в модуль підключення до Інтернету за допомогою кабелю USB. Довжина кабелю становить до 5 м, сигнали можуть спотворюватися, якщо довжина більша[31].

Потім модуль підключення до Інтернету під'єднується до міні-комп'ютера Raspberry Pi за допомогою кабелю Ethernet. Максимальна довжина кабелю без повторювачів, відповідно до стандарту, становить 100 м.





Рисунок 3.12 – Структура додавання візуального ефекту метеорологічних станцій Орегон LW301

Важливим завданням у сфері моніторингу та прогнозування комп'ютерної системи є своєчасна можливість отримання інформації про погоду. Одне з рішень цієї проблеми полягає у передачі даних з точок спостереження до системи за допомогою мобільного зв'язку. У комп'ютерній системі моніторингу та прогнозування така передача даних здійснюється через GSM-модем і спеціальний програмний модуль, який зчитує дані з датчиків і передає інформацію через модем до приймальної станції.

Тому зібрану від датчиків інформацію одноплатовий комп'ютер Raspberry Pi передає дані в Інтернет з використанням підключеного до нього USB GSM/GPRS модема з SIM-картою обраного оператора мобільного зв'язку. Виміряні дані зберігаються в Інтернеті в [www.osanywhereweather.com](http://www.osanywhereweather.com).

За допомогою цієї схеми підключення будь-якого зареєстрованого користувача, який знає параметри метеорологічних станцій, і вибравши місце на карті, можна мати віддалений доступ до метеорологічних даних зі смартфона або ПК. Потрібно завантажити відповідне програмне забезпечення для перегляду даних на телефоні– Anywhere weather.

Схема датчиків збору інформації відображається на карті обраного району (річка Дністер, село Нижнів, Тлумацький район Івано-Франківської області) (рис. 3.13). Вважаються, що потік хвиль в більшості випадків рухається набагато повільніше, ніж атмосферний фронт, тому метеорологічні датчики розташовані в точках спостереження вище по потоку річки для інтерполяції рівня води нижче за течією.

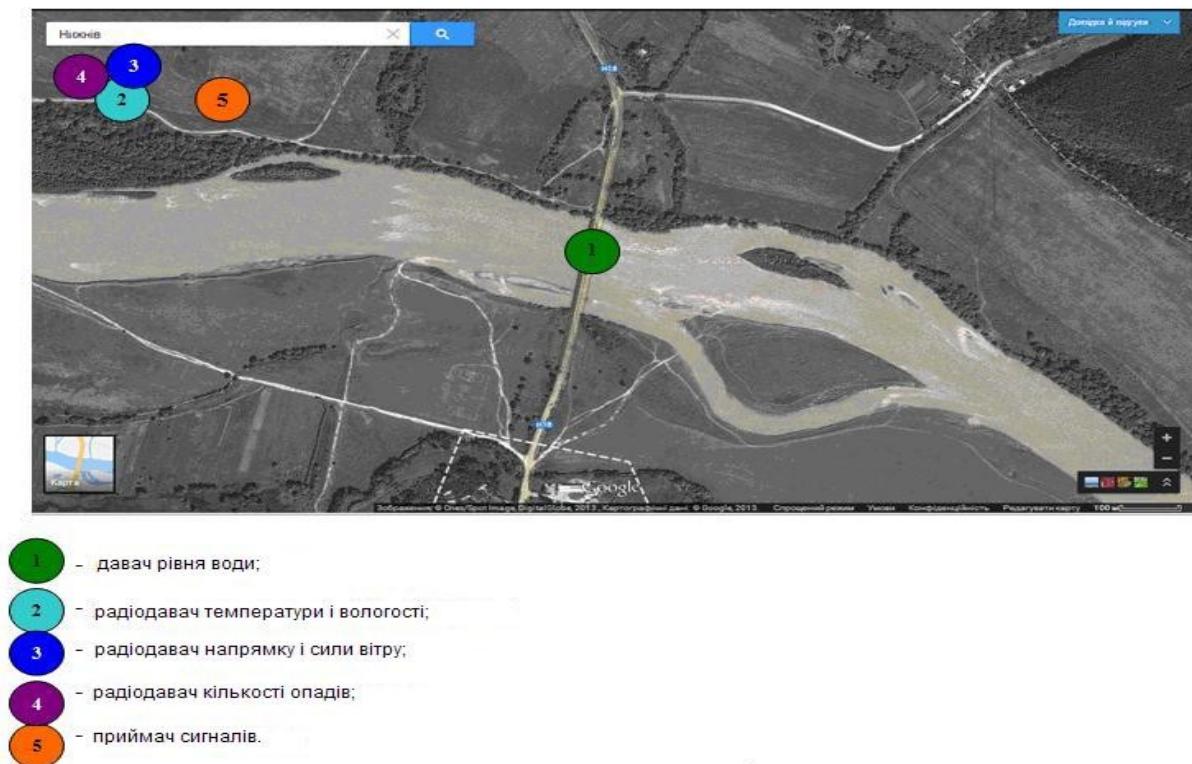


Рисунок 3.13 – Розміщення датчиків для збору даних

Отримана з точок спостереження інформація приходить на АРМ оператора (див. рис. 3.12), де проводиться обробка та зберігання інформації в базі даних. Усереднення отриманої інформації про погоду з декількох

точок спостереження здійснюється в АРМ оператора. Оскільки датчик вимірює рівень відстані від поточного рівня води, розрахунок рівня води проводиться відповідно до формули (3.1) з використанням прикладного програмного забезпечення на АРМ оператора. Візуалізація результатів здійснюється на екрані АРМ оператора як графіки зміни значень погодних умов, що значно полегшує аналіз отриманої інформації.

Прогнозування рівня води здійснюється на АРМ оператора з урахуванням погодних умов.

Автоматизовані робочі місця оператора і АРМ оператора від гідроелектростанції інтегровані в мережевий інтерфейс Ethernet за допомогою відповідного обладнання і програмного забезпечення.

### 3.4 Висновки до розділу

1. Запропоновано структуру комп'ютерної системи моніторингу та прогнозування рівнів води, технічну реалізацію системи здійснюють за допомогою системи пристроїв та відповідного програмного забезпечення, що дозволяє збирати та обробляти інформацію про погодні умови та рівень води в обраній місцевості.

2. Реалізація апаратних засобів комп'ютерної системи дозволяє віддалено виконувати збір даних у вибраних точках спостереження.

3. Розроблено спеціальне програмне забезпечення комп'ютерної системи моніторингу та прогнозування рівня води в річках, що дозволяє прогнозувати зміну рівня води в залежності від отриманих погодних параметрів.

## 4 ЕКОНОМІЧНА ЧАСТИНА

### 4.1 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки. Для проведення технологічного аудиту було залучено 2-х незалежних експертів. Такими експертами будуть Хошаба Олександр Мирославович (к.т.н., доц. кафедри ПЗ ВНТУ) та Романюк Олександр Никифорович (д.т.н., зав. Кафедри ПЗ ВНТУ).

Проведено оцінювання комерційного потенціалу за критеріями, наведеними в таблиці 5.1.

Таблиця 5.1 - Критерії оцінювання комерційного потенціалу розробки  
бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кр и- тері й	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку

Продовження таблиці 5.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-тер.	0	1	2	3	4
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту у дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
<b>Ринкові перспективи</b>					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою

Продовження таблиці 5.1

Продовження таблиці 5.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-тер.	0	1	2	3	4
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Здійснюємо оцінювання комерційного потенціалу розробки за 12-ма критеріями за 5-ти бальною шкалою.

Результати оцінювання комерційного потенціалу розробки наведено в таблиці 5.2.

Таблиця 5.2 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта	
	1. Експерт 1	2. Експерт 2
	Бали, виставлені експертами:	
1	4	4
2	3	3
3	3	4
4	4	3
5	3	4
6	4	4
7	3	3
8	4	4
9	4	3
10	4	3
11	3	4
12	3	4
Сума балів	СБ <sub>1</sub> = 43	СБ <sub>2</sub> = 43
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{2} = 43$	

Отже, з отриманих даних таблиці 5.2 видно, що нова розробка має високий рівень комерційного потенціалу.

4.2 Прогнозування витрат на виконання наукової роботи та впровадження її результатів

Для розробки нового програмного продукту необхідні такі витрати.

Основна заробітна плата для розробників визначається за формулою (5.1):

$$З_0 = \frac{М}{T_p} \cdot t, \quad (5.1)$$

де М- місячний посадовий оклад конкретного розробника;

$T_p$  - кількість робочих днів у місяці,  $T_p = 21$  день;

t - число днів роботи розробника, t = 45 днів.

Розрахунки заробітних плат для керівника і програміста наведені в таблиці 5.2.

Таблиця 5.2 – Розрахунки основної заробітної плати

Працівник	Оклад	Оплата за	Число	Витрати на
-----------	-------	-----------	-------	------------



	М, грн.	робочий день, грн.	днів роботи, t	оплату праці, грн.
Науковий керівник	5500	261,90	10	2619
Інженер-програміст	4000	190,47	45	8571,15
Всього:				11190,15

Розрахуємо додаткову заробітну плату:

$$З \text{ дод} = 0,1 \cdot 11190,15 = 1119,01 (\text{грн.})$$

Нарахування на заробітну плату операторів НЗП розраховується як 37,5...40% від суми їхньої основної та додаткової заробітної плати:

$$Н \text{ зп} = (З \text{ о} + З \text{ р}) \cdot \frac{\beta}{100}, \quad (5.2)$$

$$Н \text{ зп} = (11190,15 + 1119,01) \cdot \frac{36,3}{100} = 4468,22 (\text{грн.}).$$

Розрахунок амортизаційних витрат для програмного забезпечення виконується за такою формулою:

$$A = \frac{Ц \cdot N_a \cdot T}{100 \cdot 12}, \quad (5.3)$$

де Ц – балансова вартість обладнання, грн;

$N_a$  – річна норма амортизаційних відрахувань % (для програмного забезпечення 25%);

T – Термін використання (T=3 міс.).

Таблиця 5.3 – Розрахунок амортизаційних відрахувань

Найменування програмного забезпечення	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн
---------------------------------------	--------------------------	----------------------	---------------------------	--

Персональний комп'ютер	9000	25	2	375
Всього:				375

Розрахуємо витрати на комплектуючі. Витрати на комплектуючі розрахуємо за формулою:

$$K = \sum_{i=1}^n H_i \cdot C_i \cdot K_i, \quad (5.4)$$

де  $n$  – кількість комплектуючих;

$H_i$  - кількість комплектуючих  $i$ -го виду;

$C_i$  – покупна ціна комплектуючих  $i$ -го виду, грн;

$K_i$  – коефіцієнт транспортних витрат (приймемо  $K_i = 1,1$ ).

Таблиця 5.4 - Витрати на комплектуючі, що були використані для розробки ПЗ.

Найменування матеріалу	Одиниці виміру	Ціна, грн.	Витрачено	Вартість витрачених матеріалів, грн.
Флешка	шт.	180	1	180
Пачка паперу	уп.	130	1	130
Ручка	шт.	10	1	10
Всього з урахуванням транспортних витрат				352

Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot P \cdot \Phi \cdot K_n; \quad (5.5)$$

де  $V$  – вартість 1кВт-години електроенергії ( $V=1,7$  грн/кВт);

$\Pi$  – установлена потужність комп'ютера ( $\Pi=0,6\text{кВт}$ );

$\Phi$  – фактична кількість годин роботи комп'ютера ( $\Phi=180$  год.);

$K_n$  – коефіцієнт використання потужності ( $K_n < 1$ ,  $K_n=0,8$ ).

$$B_e = 1,7 \cdot 0,6 \cdot 180 \cdot 0,8 = 146,88 \text{ (грн.)}$$

Розрахуємо інші витрати  $B_{\text{ін}}$ .

Інші витрати  $I_b$  можна прийняти як (100...300)% від суми основної заробітної плати розробників та робітників, які були виконували дану роботу, тобто:

$$B_{\text{ін}} = (1..3) \cdot (Z_o + Z_p). \quad (5.6)$$

Отже, розрахуємо інші витрати:

$$B_{\text{ін}} = 1 \cdot (11190,15 + 1119,01) = 12309,16 \text{ (грн.)}$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини роботи:

$$B = Z_o + Z_d + H_{\text{зн}} + A + K + B_e + I_b$$

$$B = 11190,15 + 1119,01 + 4468,22 + 375 + 352 + 146,88 + 12309,16 = 29960,42 \text{ (грн.)}$$

Розрахуємо загальну вартість наукової роботи  $B_{\text{заг}}$  за формулою:

$$B_{\text{заг}} = \frac{B_{\text{ін}}}{\alpha} \quad (5.7)$$

де  $\alpha$  – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях = 1.

$$B_{\text{заг}} = \frac{29960,42}{1} = 29960,42$$

Прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою:

$$ЗВ = \frac{B_{заг}}{\beta} \quad (5.8)$$

де  $\beta$  – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Отже, розрахуємо загальні витрати:

$$ЗВ = \frac{29960,42}{0,9} = 33289,35 \text{ (грн.)}$$

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

Спрогнозуємо отримання прибутку від реалізації результатів нашої розробки. Зростання чистого прибутку можна оцінити у теперішній вартості грошей. Це забезпечить підприємству (організації) надходження додаткових коштів, які дозволять покращити фінансові результати діяльності .

Оцінка зростання чистого прибутку підприємства від впровадження результатів наукової розробки. У цьому випадку збільшення чистого прибутку підприємства  $\Delta \Pi_i$  для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta \Pi_i = \sum_1^n \dot{\dot{i}} \quad (5.9)$$

де  $\Delta \Pi_{я}$  – покращення основного якісного показника від впровадження результатів розробки у даному році;

$N$  – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

$\Delta N$  – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$P_y$  – основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

$n$  – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

В результаті впровадження результатів наукової розробки витрати на виготовлення інформаційної технології зменшаться на 20 грн (що автоматично спричинить збільшення чистого прибутку підприємства на 20 грн), а кількість користувачів, які будуть користуватись збільшиться: протягом першого року – на 250 користувачів, протягом другого року – на 200 користувачів, протягом третього року – 150 користувачів. Реалізація інформаційної технології до впровадження результатів наукової розробки складала 600 користувачів, а прибуток, що отримував розробник до впровадження результатів наукової розробки – 300 грн.

Спрогнозуємо збільшення чистого прибутку від впровадження результатів наукової розробки у кожному році відносно базового.

Отже, збільшення чистого продукту  $\Delta P_1$  протягом першого року складатиме:

$$\Delta P_1 = 20 \cdot 600 + (300 + 20) \cdot 250 = 92000 \text{ грн.}$$

Протягом другого року:

$$\Delta P_2 = 20 \cdot 600 + (300 + 20) \cdot (250 + 200) = 156000 \text{ грн.}$$

Протягом третього року:

$$\Delta P_3 = 20 \cdot 600 + (300 + 20) \cdot (250 + 200 + 150) = 204000 \text{ грн.}$$

#### 4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Визначимо абсолютну і відносну ефективність вкладених інвестором інвестицій та розрахуємо термін окупності.

Абсолютна ефективність  $E_{abc}$  вкладених інвестицій розраховується за формулою:

$$E_{abc} = (ПП - PV), \quad (5.10)$$

де  $\Delta\Pi_i$  – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

$t$  – період часу, протягом якого виявляються результати впровадженої НДДКР, 3 роки;

$r$  – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

$t$  – період часу (в роках) від моменту отримання чистого прибутку до точки 2, 3, 4.

Рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, рисунок 5.1.

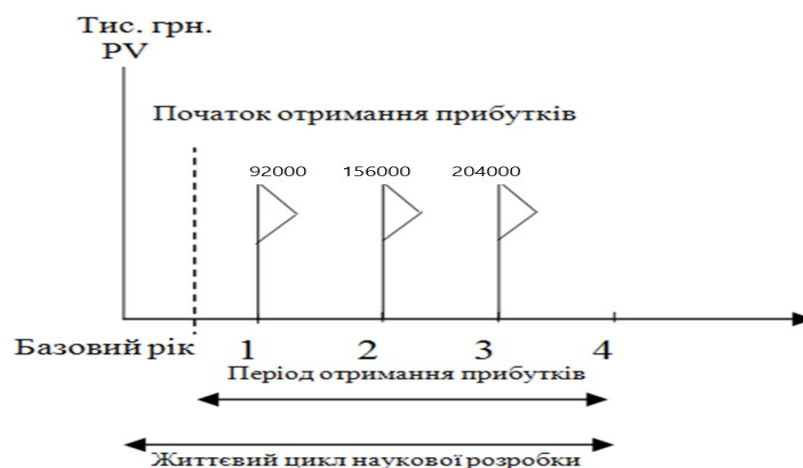


Рисунок 5.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

Розрахуємо вартість чистих прибутків за формулою:

$$ПП = \sum_1^m \frac{\Delta \Pi_i}{i^t} \quad (5.11)$$

де  $\Delta \Pi_i$  – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

$t$  – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

$i$  – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

$t$  – період часу (в роках) від моменту отримання чистого прибутку до точки.

Отже, розрахуємо вартість чистого прибутку:

$$ПП = \frac{33289,35}{i^t} \text{ (грн.)}$$

Тоді розрахуємо  $E_{abc}$ :

$$E_{abc} = 365863,9 - 33289,35 = 332574,55 \text{ грн.}$$

Оскільки  $E_{abc} > 0$ , то вкладання коштів на виконання та впровадження результатів НДДКР буде доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій  $E_v$  за формулою:

$$E_v = \sqrt[T]{1 + \frac{E_{abc}}{PV}} - 1 \quad (5.12)$$

де  $E_{abc}$  – абсолютна ефективність вкладених інвестицій, грн;

$PV$  – теперішня вартість інвестицій  $PV = 3B$ , грн;

$T_j$  – життєвий цикл наукової розробки, роки.

Тоді будемо мати:

$$E_v = \sqrt[3]{1 + \frac{332574,55}{33289,35}} - 1 = 1,22 \text{ або } 122\%$$

Далі, розраховану величина  $E_v$  порівнюємо з мінімальною (бар'єрною) ставкою дисконтування  $\tau_{\text{мін}}$ , яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування  $\tau_{\text{мін}}$  визначається за формулою:

$$\tau = d + f,$$

де  $d$  – середньозважена ставка за депозитними операціями в комерційних банках; в 2019 році в Україні  $d = 0,2$ ;

$f$  – показник, що характеризує ризикованість вкладень, величина  $f = 0,1$ .

$$\tau = 0,2 + 0,1 = 0,3$$

Оскільки  $E_v = 122\% > \tau_{\text{мін}} = 0,3 = 30\%$ , то у інвестор буде зацікавлений вкладати гроші в дану наукову розробку.



Термін окупності вкладених у реалізацію наукового проекту інвестицій. Термін окупності вкладених у реалізацію наукового проекту інвестицій  $T_{ок}$  розраховується за формулою:

$$T_{ок} = \frac{1}{E_e}$$

$$T_{ок} = \frac{1}{1,22} = 0,81 \text{ року}$$

Обрахувавши термін окупності даної наукової розробки, можна зробити висновок, що фінансування даної наукової розробки буде доцільним.

#### 4.5 Висновки до розділу

Проведено оцінювання комерційного потенціалу розробки із залученням незалежних експертів. За результатами оцінки проект показав комерційний потенціал вище середнього.

Виконано прогнозування витрат на виконання на наукової роботи та впровадження її результатів. Загальні витрати на виконання та впровадження результатів наукової розробки дорівнюють 33289,35 грн.

Виконано прогнозування комерційних ефектів від реалізації результатів розробки. За результатами розрахунків, збільшення чистого прибутку підприємства рівне 92 тисяч грн, 156 тисяч грн, 204 тисяч грн за перший, другий і третій рік після впровадження результатів наукової розробки відповідно.

Розраховано ефективність вкладених інвестицій та періоду їх окупності. Абсолютна ефективність вкладених інвестицій рівна 332574,55 гривень; відносна ефективність вкладених інвестицій рівна 122%; термін окупності – 0,81 року, що підтверджує доцільність розробки та впровадження на ринок програмного продукту.



## ВИСНОВКИ

У магістерській кваліфікаційній роботі проведено детальний аналіз методів оптимізації комп'ютерних систем прогнозування рівня води річок. Сформульовано мету досліджень - підвищення рівня прогнозування рівня води річок за рахунок розробка моделі та методу оптимізації.

Запропоновано математичну модель екологічної поведінки річки, у якій, на відміну від відомої, враховано коливальний процес та вагові коефіцієнти гармонійних складових, що дозволило підвищити точність моделі.

Запропоновано метод оптимізації прогнозування рівня води річок, який відрізняється від відомого використанням нейромереж, що дозволило підвищити достовірність оптимізації та комп'ютеризувати його.

Розроблено програмний додаток для підвищеної точності прогнозування рівня води річок.

Отримані в магістерській кваліфікаційній роботі наукові та практичні результати можна використати для прогнозування рівня води річок.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Гусак М.А. Програмні методи оптимізації комп'ютерної системи прогнозування рівня води річок Прикарпаття/ М.А. Гусак – В.: ВНТУ, 2019.
2. Иващук О. А. Теоретические основы построения автоматизированной системы управления экологической безопасностью промышленно–транспортного комплекса: монография / О. А. Иващук, И. С. Константинов. – М: Машиностроение, 2009. – 205с.
3. Иващук О. А. Автоматизация как основа реализации принципов современной системы управления экологической безопасностью / О. А. Иващук // Информационные системы и технологии. ИзвестияОрелГТУ.– 2009. – № 4/54(565) – С. 95 – 104.
4. Мельников А. В. Общая классификация и характеристика экологических систем управления / А. В. Мельников, В. Н. Мельников // Вестн. Астрахан. гос. техн. ун–та. Сер.: Рыбное хозяйство. – 2009. – № 1. – С. 48–52.
5. Sroufe Robert. "Effects of Environmental Management Systems on Environmental Management Practices and Operations." Production and Operations Management. 12–3 (2003):416–431.
6. Иващук О. А. Модель объекта управления АСУ экологической безопасностью территории жилой застройки / О. А. Иващук, О. Д. Иващук // ИТНОП – 2012. – 5с.
7. Синеглазов В. М. Автоматизация технологических процессов: навчальний посібник / В. М. Синеглазов, І. Ю. Сергеев. – К.: НАУ, 2010. – 506с.
8. Проектирование систем автоматизации технологических процессов: Справочное пособие / А. С. Клюев, Б. В. Глазов, А. Х. Дубровский, А. А. Клюев; под ред. А. С. Клюева. – 2–е изд., перераб. и доп. – М.: Энергоатомиздат, 1990. – 464с.
9. Leigh E. On the relation between the productivity, biomass, diversity and

- stability of a community / E. Leigh // Proc. Natl. Acad. Sci. U. S. – 1965. – № 53. – P. 777–783.
10. Leigh E. The ecological role of Volterra's equations. In Some Mathematical Problems in Biology / E. Leigh. – M. Gerstenhaber, ed., Providence: The American Mathematical Society, 1968.
  11. Ляпунов А. М. Избранные труды; ред. акад. А. И. Смирнова / А. М. Ляпунов. – М.: Изд. АН СССР, 1948. – 540с.
  12. Самарский А. А. Математическое моделирование / А. А. Самарский, А. П. Михайлов. – М.: Наука, 2005. – 320с.
  13. Форрестер Дж. Мировая динамика / Дж. Форрестер; пер. с англ. А. Ворощука и С. Иегова по ред. Д. Гвишиани и Н. Моисеева. – М.: ООО «Издательство АСТ»; СПб.: Terra Fantastica, 2003. – 379 с.
  14. Лаврик В. І. Моделювання і прогнозування стану довкілля: підручник / В. І. Лаврик. – К.: Академія, 2010. – 400с.
  15. Ковальчук П. І. Моделювання і прогнозування стану навколишнього середовища: навч. посібник / П. І. Ковальчук. – К.: Либідь, 2003. – 208с.
  16. Левич А. П. Математические аспекты вариационного моделирования в экологии сообществ / А. П. Левич, В. Л. Алексеев, В. А. Никулин // Математическое моделирование. – 1994. – № 5. – С. 55–76.
  17. Ляшенко І. М. Основи математичного моделювання економічних, екологічних та соціальних процесів: навчальний посібник / І. М. Ляшенко, М. В. Коробова, А. М. Столяр. – Тернопіль: Навчальна книга – Богдан, 2006. – 304с.
  18. Петренко В. В. Картографічне моделювання геохімічних полів / В. В. Петренко // Ученые записки Таврического национального университета им. В. И. Вернадского. Серия «География». – 2011. – Т. 24 (63). – № 3. – С. 144–151.
  19. Сидоренко Ю. В. Модифікований спосіб симплексної вагової інтерполяції функції багатьох змінних [Електронний ресурс] / Ю. В. Сидоренко, О. С. Каленюк // Прикладна геометрія та інженерна графіка.

- 2009. – № 44. – 6 с. – Режим доступу:  
[http://archive.nbu.gov.ua/portal/chem\\_biol/PTdau/2009\\_4\\_44/1/16.pdf](http://archive.nbu.gov.ua/portal/chem_biol/PTdau/2009_4_44/1/16.pdf).
20. Сукач Ю. Г. Основи прогнозування повеней [Електронний ресурс] / Ю. Г. Сукач, О. С. Бабаджанова // Львівський державний університет безпеки життєдіяльності, м. Львів – 2 с. – Режим доступу:<http://ubdg.lviv.ua>.
  21. Определение расчетных гидрологических характеристик: СНиП 2.01.14–83 [Електронний ресурс]. – Режим доступу:[budstandart.ua](http://budstandart.ua).
  22. Пособие по определению расчетных гидрологических характеристик. – Л.: Гидрометеиздат, 1984. – 448с.
  23. Визначення розрахункових гідрологічних характеристик: ДБН В.2.4 – Х:201Х. – Офіц. вид. – К.: М–во регіонального розвитку, будівництва та житлово–комунального господарства, 2012. – 102с.
  24. Кендзіор З. Автоматизована система моніторингу рівня забруднення рік та керування процесами очищення: автореф. дис. канд. техн. наук: спец. 05.13.06 “Автоматизовані системи управління та прогресивні інформаційні технології” / З. Кендзіор. – Л., 2004. – 20 с.
  25. Акіменко В. В. Моделі й системи аналізу і прогнозування техногенного забруднення приграничного шару атмосфери: автореф. дис. д–ра. техн. наук: спец. 01.05.04 “Системний аналіз і теорія оптимальних рішень” / В. В. Акіменко. – К., 2001. – 34с
  26. Иващук О. А. Автоматизация и интеллектуализация управления экологической ситуацией на урбанизированных территориях / О. А. Иващук, О. Д. Иващук // Научно–технический журнал Госуниверситет – УНПК. – 2013.– №6(80) – С.61–65.
  27. Методика дослідження екологічного стану техногенно– трансформованих ґрунтів на основі теорії нейромереж / О. В. Пендерецький, Г. В. Кіт, М. І. Горбійчук, М. А. Шуфнарівич // Екологічні проблеми техногенного– навантажених регіонів: матеріали міжнародної науково–практичної конференції, м. Дніпропетровськ, травень 2008р. – Дніпропетровськ, 2008. – С. 191–192.

28. Ермаков С. М. Математическая теория оптимального эксперимента / С. М. Ермаков, А. А. Жиглявский. – М.: Наука, 1987. – 320с.
29. Справочник по типовым программам моделирования / А. Г. Ивахненко, Ю. В. Коппа, В. С. Степашко и др.; под ред. А. Г. Ивахненко. – К.: Техника, 1980. – 184с.
30. Круглов В. В. Искусственные нейронные сети. Теория и практика / В. В. Круглов, В. В. Борисов. – М.: Горячая линия – Телеком, 2001. – 382с.
31. Осовский С. Нейронные сети для обработки информации: пер. с польского / С. Осовский. – М.: Финансы и статистика, 2004. – 344с.

# ДОДАТКИ



## Додаток А. Технічне завдання

Технічне завдання на магістерську кваліфікаційну роботу  
Міністерство освіти і науки України  
Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ  
д.т.н., проф. О.Н. Романюк  
" \_\_\_\_ " \_\_\_\_\_ 2019р.

Технічне завдання  
на магістерську кваліфікаційну роботу за спеціальністю  
121 – Інженерія програмного забезпечення

Керівник магістерської кваліфікаційної роботи:  
\_\_\_\_\_ д.т.н, проф. А.М. Пестух, д.т.н., проф. О.Н. Романюк  
" \_\_\_\_ " \_\_\_\_\_ 2019 р.

Виконав:  
\_\_\_\_\_ студент гр.1ПІ-18м М.А. Гусак  
" \_\_\_\_ " \_\_\_\_\_ 2019 р.

Вінниця – 2019 року

### 1. Найменування та галузь застосування

Магістерська кваліфікаційна робота: «Розробка програмних методів оптимізації комп'ютерної системи прогнозування рівня води річок Прикарпаття».

Галузь застосування – комп'ютерні системи для прогнозування рівня води річок.

### 2. Підстава для розробки.

Підставою для розробки магістерської кваліфікаційної роботи є рішення засідання кафедри програмного забезпечення (протокол №\_\_ від «\_\_»\_\_\_\_\_ 2019 року).

### 3. Мета та призначення розробки.

Метою роботи є підвищення ефективності прогнозування рівня води річок за рахунок розробки моделі та методу оптимізації.

Призначення роботи – розробка методів оптимізації комп'ютерної системи прогнозування рівня води річок.

### 4. Джерела розробки:

– завдання на магістерську кваліфікаційну роботу.

### 5. Технічні вимоги

Тип річок – річки Прикарпаття; алгоритм опису процесів з некротними частотами – на основі генетичних алгоритмів; прогнозування стану річки – за спостереженнями; поточні дані про параметри погодних умов: температура, вологість, атмосферний тиск, опади, напрямок вітру та дані рівня води; мова програмування - C#.

### 6. Конструктивні вимоги.

Комп'ютерну систему реалізувати з використання засобів обчислювальної техніки та датчиків KROHNEOPTIWAVE 5200 C/F.

Графічна та текстова документація повинна відповідати діючим стандартам України.

7. Перелік технічної документації, що пред'являється по закінченню робіт:

- пояснювальна записка до МКР;
- технічне завдання;
- лістинги програми.

8. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ.

9. Стадії та етапи розробки:

№ з/п	Назва етапів магістерської кваліфікаційної Роботи	Строк виконання етапів роботи	Примітка
1	Аналіз існуючих методів оптимізації комп'ютерної системи прогнозування рівня води річок	23.09.2019 – 03.10-2019	Вик.
2	Розробка математичної моделі на основі генетичних алгоритмів	03.10.2019 – 21.10.2019	Вик.
3	Розробка програмного методу оптимізації комп'ютерної системи прогнозування рівня води річок	21.10.2019 – 10.11.2019	Вик.
4	Економічна частина	10.11.2019 – 24.11.2019	Вик.

10. Порядок контролю та прийняття.

Виконання етапів магістерської кваліфікаційної роботи контролюється керівником згідно з графіком виконання роботи.

Прийняття магістерської кваліфікаційної роботи здійснюється ДЕК, затвердженою зав. кафедрою згідно з графіком захисту.

## Додаток Б. Ілюстративний матеріал

ІЛЮСТРАТИВНИЙ МАТЕРІАЛ ДО ЗАХИСТУ МАГІСТЕРСЬКОЇ  
КВАЛІФІКАЦІЙНОЇ РОБОТИ

Завідувач кафедри ПЗ, д. т. н., професор \_\_\_\_\_ О.Н. Романюк

Науковий керівник, д. т. н., проф кафедри ПЗ \_\_\_\_\_ О. Н. Романюк

Рецензент, к.т.н, доцент кафедри ПЗ \_\_\_\_\_ В.В. Колодний

Нормоконтроль, д. т. н., проф. кафедри ПЗ \_\_\_\_\_ О. Н. Романюк

Виконавець, студент групи 1ПІ-18м \_\_\_\_\_ М.А. Гусак

Плакат 1 – Тема, автор, науковий керівник



«Розробка програмних методів  
оптимізації комп'ютерної системи  
прогнозування рівня води річок  
Прикарпаття»

---

Виконав студент  
групи 1ПІ- 18м  
Гусак Михайло

Науковий керівник  
зав.кафедри ПЗ, д.  
т.н., проф.  
Романюк О.Н.

## Плакат 2 – Актуальність



### Актуальність

Активізація в останні роки в Україні повеней в силу різних природних і антропогенних процесів є дійсно загрозою – постає питання прогнозування і моделювання їх виникнення.

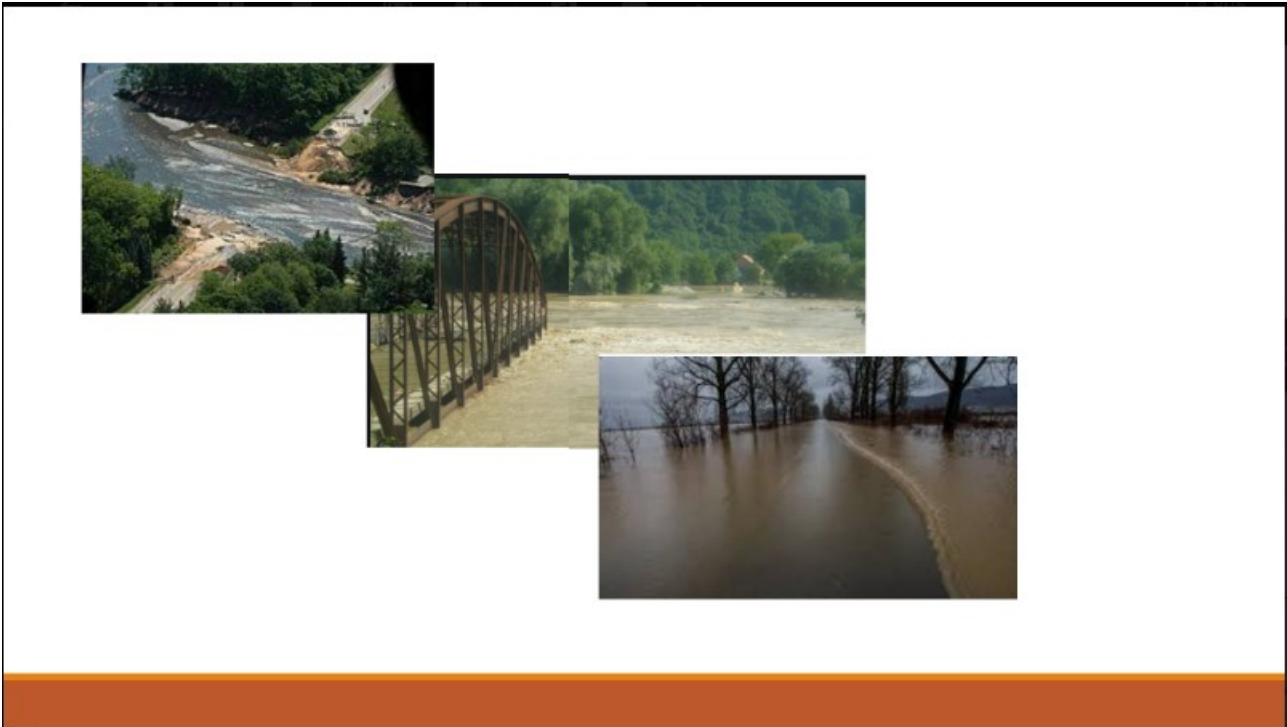
За останні 20 років на території України мали місце значні паводки, що призвели до виникнення надзвичайних ситуацій. Так, середньорічні збитки від паводків у 2018 роках становили близько 6 млрд гривень.



Прикарпатські річки характеризуються раповим, стихійним проходженням паводків, які часто призводять до катастрофічних наслідків та можуть бути небезпечними для життєдіяльності людини.

Екологічне прогнозування річок – це передбачення можливої поведінки природних систем річки, що визначається природними процесами. Головна мета прогнозу – оцінка передбачуваної реакції довкілля на пряму або опосередковану дію поведінки річки.

Плакат 3 – Наслідки руйнівної сили повеней на Прикарпатті





## Плакат 4 - Мета, об'єкт та предмет дослідження

### Мета, предмет, об'єкт, задачі:



**Метою магістерської** кваліфікаційної роботи є підвищення рівня прогнозування рівня води річок за рахунок розробки моделі та методу оптимізації.

**Об'єкт дослідження** – процес прогнозування рівня води річок

**Предмет дослідження** – методи та засоби прогнозування рівня води річок.

**Основними задачами дослідження є:**

провести аналіз існуючих методів і засобів аналізу рівня води річок;

розробка моделі екологічної поведінки річки, у якій на відміну від відомої враховано коливальний процес та вагові коефіцієнти гармонійних складових, що дозволить підвищити точність моделі;

розробка методу оптимізації прогнозування рівня води річок Прикарпаття, який відрізняється від відомого використанням нейромереж, що дозволило підвищити достовірність оптимізації та комп'ютеризувати його;

## Плакат 5 - Наукова новизна отриманих результатів

### Наукова новизна отриманих результатів



Вперше розроблено математичну модель екологічної поведінки річки, у якій, на відміну від відомої, враховано коливальний процес та вагові коефіцієнти гармонійних складових, що дозволило підвищити точність моделі.

Подальшого розвитку отримав метод оптимізації прогнозування рівня води річок, який відрізняється від відомого використанням нейромереж, що дозволило підвищити достовірність оптимізації та комп'ютеризувати його.

## Плакат 6 - Практична цінність



### Практична цінність

---

Практична цінність полягає в наступному:

на основі отриманих у магістерській кваліфікаційній роботі теоретичних положень запропоновано алгоритми та розроблено програмні засоби для прогнозування рівня річок.

## Плакат 7 - Математична модель екологічної поведінки річки

Екологічний процес, може бути поданий у вигляді математичної моделі, що містить лінійну тенденцію  $h(t)$ , гармонічну складову  $G(t)$  і залишкову складову  $H_t$ , а саме:

$$\tilde{H}_t = H_t + G(t) + h(t)$$

Гармонічна складова з некрратними частотами, що притаманна досліджуваним процесам, також процесам з екологічним характером, моделюється таким виразом

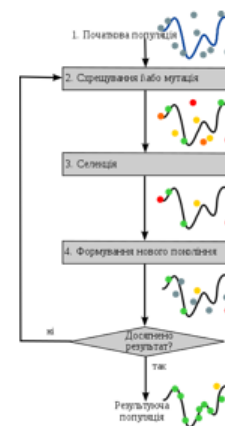
$$G(t) = A_0 + \sum_{j=1}^n (A_j \sin(t\omega_j) + B_j \cos(t\omega_j)),$$

де  $t$  – такти відліку часу,  $t = 1, 2, 3, \dots, N$ ,

## Плакат 8 - Метод оптимізації прогнозування рівня води

### Використання нейромережі

Випадковим чином в масиві створюється деяка кількість початкових елементів «осіб», або початкова популяція. Особи оцінюються з використанням функції пристосування, в результаті якої кожній особі присвоюється певне значення пристосованості, яке визначає можливість виживання особи. Після цього з використанням отриманих значень пристосованості вибираються особи, допущені до схрещення. До осіб застосовується «генетичні оператори», створюючи таким чином наступне покоління осіб. Особи наступного покоління також оцінюються застосуванням генетичних операторів і виконується селекція і мутація. Так моделюється еволюційний процес поведження річки ,



Подальшого розвитку отримав метод оптимізації прогнозування рівня води річок, який відрізняється від відомого використанням нейромереж, що дозволило підвищити достовірність оптимізації та кваліфікувати його.

## Плакат 9 - Складові генетичного алгоритму



### Складові генетичного алгоритму

Розроблена комп'ютерна система виконує функцію моніторингу і прогнозування. Вона інтегрує підсистеми збору і обробки інформації про погодні умови та рівень води в річках і підсистем аналізу та прогнозування рівнів води на основі даних.

Завдання, які вирішуються за допомогою комп'ютерної системи:

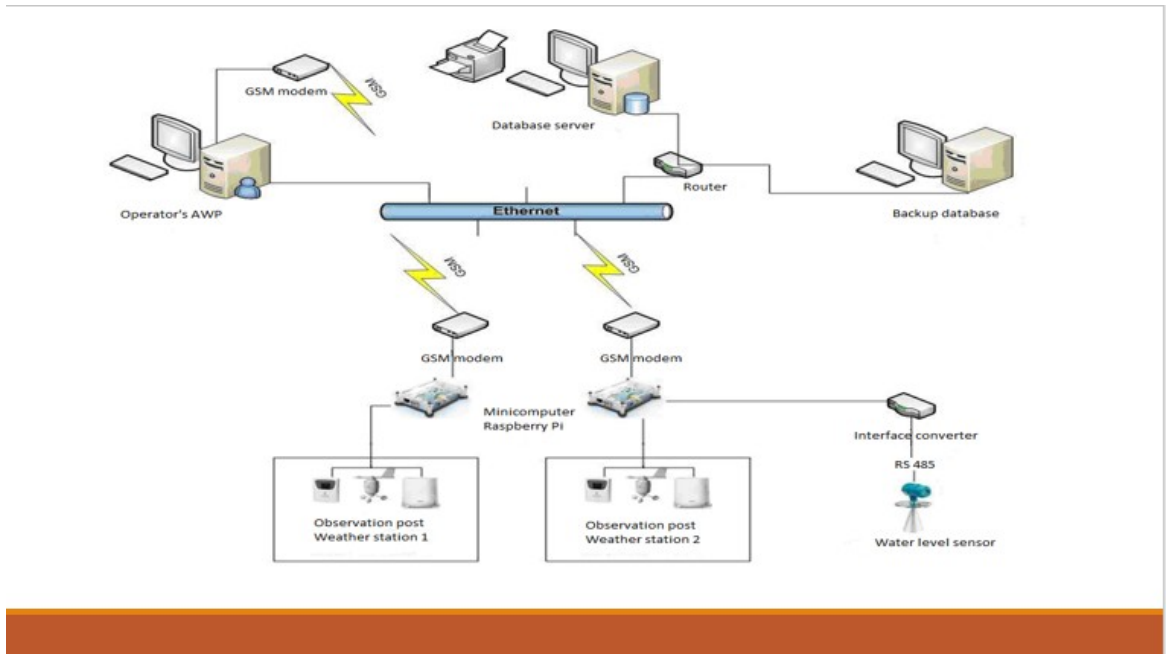
дослідження впливу погодних умов на рівень води в річках за допомогою дистанційного моніторингу;

аналіз і оцінювання річкового стоку води;

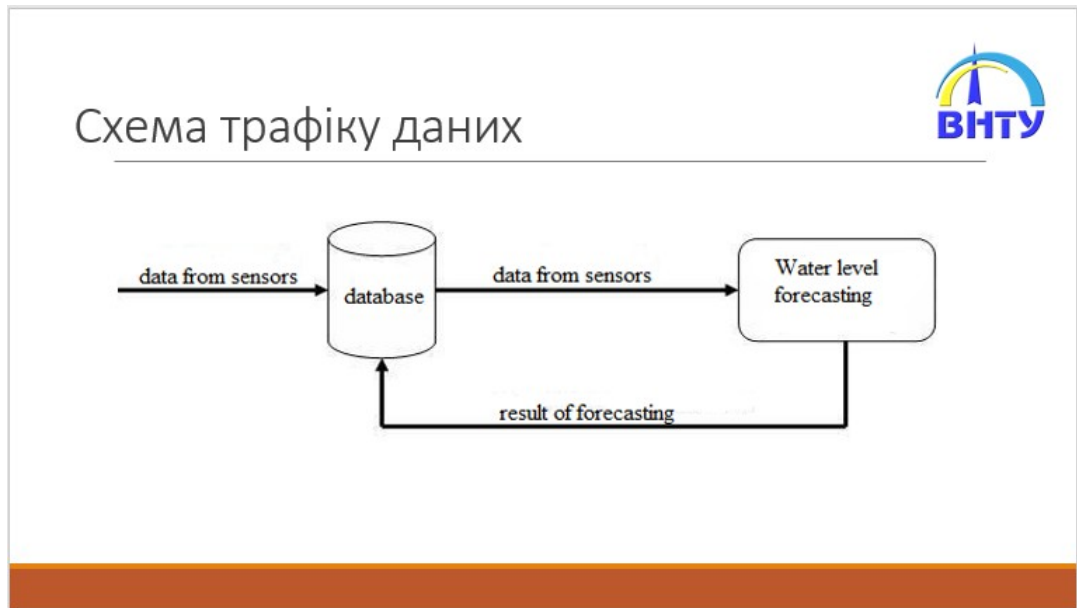
прогнозування стану річки (за спостереженнями);

управління повеннями.

## Плакат 10 - Комп'ютерна система прогнозування рівня води



Плакат 11 – Схема трафіку даних





## Плакат 12 – Висновки

---

### Висновки



Проведено детальний аналіз методів оптимізації комп'ютерних систем прогнозування рівня води річок. Сформульовано мету досліджень - підвищення рівня прогнозування рівня води річок за рахунок розробка моделі та методу оптимізації.

Запропоновано математичну модель екологічної поведінки річки, у якій, на відміну від відомої, враховано коливальний процес та вагові коефіцієнти гармонійних складових, що дозволило підвищити точність моделі.

Запропоновано метод оптимізації прогнозування рівня води річок, який відрізняється від відомого використанням нейромереж, що дозволило підвищити достовірність оптимізації та комп'ютеризувати його.

Розроблено програмний додаток для підвищеної точності прогнозування рівня води річок.

---

Плакат 13 – Дякую за увагу!

---

ДЯКУЮ ЗА УВАГУ!

---



## Додаток В. Лістинг головного модуля програми

```

using MyMeasurment.Models;
using System.Net.Http;
using System.Threading.Tasks;
using static Newtonsoft.Json.JsonConvert;

namespace MyMeasurment
{
    public enum Directions
    {
        South,
        West,
        East,
        North,

        SouthEast,
        SouthWest,
        NorthEast,
        NorthWest
    }

    public class WeatherService
    {
        const string WeatherCoordinatesUri = "http://api.openweathermap.org/data/
2.5/weather?
lat={0}&lon={1}&units={2}&appid=fc9f6c524fc093759cd28d41fda89a1b";
        const string WeatherCityUri =
"http://api.openweathermap.org/data/2.5/weather?
q={0}&units={1}&appid=fc9f6c524fc093759cd28d41fda89a1b";
        const string ForecaseUri =
"http://api.openweathermap.org/data/2.5/forecast?
id={0}&units={1}&appid=fc9f6c524fc093759cd28d41fda89a1b";

        public async Task<WeatherRoot>GetWeather(double latitude, double
longitude, Units units = Units.Imperial)
        {
            using (var client = new HttpClient())
            {
                var url = string.Format(WeatherCoordinatesUri, latitude, longitude,
units.ToString().ToLower());
                var json = await client.GetStringAsync(url);

                if (string.IsNullOrWhiteSpace(json))
                    return null;
            }
        }
    }
}

```

```

        return DeserializeObject<WeatherRoot>(json);
    }
}

public async Task<WeatherRoot>GetWeather(string city, Units units =
Units.Imperial)
{
    using (var client = new HttpClient())
    {
        var url = string.Format(WeatherCityUri, city,
units.ToString().ToLower());
        var json = await client.GetStringAsync(url);

        if (string.IsNullOrEmpty(json))
            return null;

        return DeserializeObject<WeatherRoot>(json);
    }
}

public async Task<WeatherForecastRoot>GetForecast(int id, Units units =
Units.Imperial)
{
    using (var client = new HttpClient())
    {
        var url = string.Format(ForecaseUri, id, units.ToString().ToLower());
        var json = await client.GetStringAsync(url);

        if (string.IsNullOrEmpty(json))
            return null;

        return DeserializeObject<WeatherForecastRoot>(json);
    }
}
}
}
namespace MyWeather.Models
{
    public class Coord
    {
        [JsonProperty("lon")]

```

```

    public double Longitude { get; set; } = 0;

    [JsonProperty("lat")]
    public double Latitude { get; set; } = 0;
}

public class Sys
{

    [JsonProperty("country")]
    public string Country { get; set; } = string.Empty;
}

public class Weather
{
    [JsonProperty("id")]
    public int Id { get; set; } = 0;

    [JsonProperty("main")]
    public string Main { get; set; } = string.Empty;

    [JsonProperty("description")]
    public string Description { get; set; } = string.Empty;

    [JsonProperty("icon")]
    public string Icon { get; set; } = string.Empty;
}

public class Main
{
    [JsonProperty("temp")]
    public double Temperature { get; set; } = 0;
    [JsonProperty("pressure")]
    public double Pressure { get; set; } = 0;

    [JsonProperty("humidity")]
    public double Humidity { get; set; } = 0;
    [JsonProperty("temp_min")]
    public double MinTemperature { get; set; } = 0;

    [JsonProperty("temp_max")]
    public double MaxTemperature { get; set; } = 0;
}

public class Wind

```

```

{
    [JsonProperty("speed")]
    public double Speed { get; set; } = 0;

    [JsonProperty("deg")]
    public double WindDirectionDegrees { get; set; } = 0;
}

public class Clouds
{
    [JsonProperty("all")]
    public int CloudinessPercent { get; set; } = 0;
}

public class WeatherRoot
{
    [JsonProperty("coord")]
    public Coord Coordinates { get; set; } = new Coord();

    [JsonProperty("sys")]
    public Sys System { get; set; } = new Sys();

    [JsonProperty("weather")]
    public List<Weather> Weather { get; set; } = new List<Weather>();

    [JsonProperty("main")]
    public Main MainWeather { get; set; } = new Main();

    [JsonProperty("wind")]
    public Wind Wind { get; set; } = new Wind();

    [JsonProperty("clouds")]
    public Clouds Clouds { get; set; } = new Clouds();

    [JsonProperty("id")]
    public int CityId { get; set; } = 0;

    [JsonProperty("name")]
    public string Name { get; set; } = string.Empty;

    [JsonProperty("dt_txt")]
    public string Date { get; set; } = string.Empty;
}

```

```

    [JsonIgnore]
    public string DisplayDate
=>DateTime.Parse(Date).ToLocalTime().ToString("g");
    [JsonIgnore]
    public string DisplayTemp => $"Temp: {MainWeather?.Temperature ?? 0}°
{Weather?[0]?.Main ?? string.Empty}";
    [JsonIgnore]
    public string DisplayIcon =>
$"http://openweathermap.org/img/w/{Weather?[0]?.Icon}.png";
}

public class WeatherForecastLevel
{
    [JsonProperty("city")]
    public City City { get; set; }
    [JsonProperty("cod")]
    public string Vod { get; set; }
    [JsonProperty("message")]
    public double Message { get; set; }
    [JsonProperty("cnt")]
    public int Cnt { get; set; }
    [JsonProperty("list")]
    public List<WeatherRoot> Items { get; set; }
}

public class River
{
    [JsonProperty("id")]
    public int Id { get; set; }
    [JsonProperty("name")]
    public string Name { get; set; }
    [JsonProperty("coord")]
    public Coord Coord { get; set; }
    [JsonProperty("country")]
    public string Country { get; set; }
    [JsonProperty("population")]
    public int Population { get; set; }
    [JsonProperty("sys")]
    public Sys Sys { get; set; }
}
}
namespace MyWeather.ViewModels
{

```

```

public class WeatherViewModel : BaseViewModel
{
    WeatherServiceWeatherService { get; } = new WeatherService();

    string location = Settings.City;
    public string Location
    {
        get { return location; }
        set
        {
            SetProperty(ref location, value);
            Settings.City = value;
        }
    }

    bool useGPS;
    public bool UseGPS
    {
        get { return useGPS; }
        set
        {
            SetProperty(ref useGPS, value);
        }
    }

    bool isImperial = Settings.IsImperial;
    public bool IsImperial
    {
        get { return isImperial; }
        set
        {
            SetProperty(ref isImperial, value);
            Settings.IsImperial = value;
        }
    }

    string temp = string.Empty;
    public string Temp
    {
        get { return temp; }
        set { SetProperty(ref temp, value); }
    }

    string condition = string.Empty;

```



```

public string Condition
{
    get { return condition; }
    set { SetProperty(ref condition, value); ; }
}

```

```

WeatherForecastRoot forecast;
public static Dictionary<string, int>
{
    get { return forecast; }
    set { forecast = value; OnPropertyChanged(); }
}

```

```

ICommand GetData();
    public void RequestAndSaveSensorsData =>
getWeather ??
                (getWeather = new Command(async () => await
ExecuteGetWeatherCommand()));

```

```

private async Task ExecuteGetWeatherCommand()
{
    if (IsBusy)
        return;
}

```

```

IsBusy = true;
    try
    {
WeatherRoot weatherRoot = null;
        var units = IsImperial ? Units.Imperial : Units.Metric;

```

```

    if (UseGPS)
    {
        var hasPermission = await CheckPermissions();
        if (!hasPermission)
            return;

```

```

        var position = await Geolocation.GetLastKnownLocationAsync();

```

```

    if (position == null)
    {
        // get full location if not cached.

```

```

        position = await Geolocation.GetLocationAsync(new
GeolocationRequest
    {
DesiredAccuracy = GeolocationAccuracy.Medium,
        Timeout = TimeSpan.FromSeconds(30)
    });
    }

weatherRoot = await WeatherService.GetWeather(position.Latitude,
position.Longitude, units);
    }
    else
    {
        //Get weather by city
weatherRoot = await WeatherService.GetWeather(Location.Trim(), units);
    }

    //Get forecast based on cityId
        Forecast = await WeatherService.GetForecast(weatherRoot.CityId,
units);

    var unit = IsImperial ? "F" : "C";
        Temp = $"Temp: {weatherRoot?.MainWeather?.Temperature ??
0}°{unit}";
        Condition = $"{weatherRoot.Name}: {weatherRoot?.Weather?
[0]?.Description ?? string.Empty}";

        await TextToSpeech.SpeakAsync(Temp + " " + Condition);
    }
    catch (Exception ex)
    {
        Temp = "Unable to get Weather";
System.Diagnostics.Debug.WriteLine(ex.Message);
    }
    finally
    {
IsBusy = false;
    }
    }

    async Task<bool>CheckPermissions()
    {
        var permissionStatus = await
CrossPermissions.Current.CheckPermissionStatusAsync(Permission.Location);

```

```

bool request = false;
if (permissionStatus == PermissionStatus.Denied)
{
    if (Device.RuntimePlatform == Device.iOS)
    {

        var title = "Location Permission";
        var question = "To get your current city the location permission is
required. Please go into Settings and turn on Location for the app.";
        var positive = "Settings";
        var negative = "Maybe Later";
        var task = Application.Current?.MainPage?.DisplayAlert(title,
question, positive, negative);
        if (task == null)
            return false;

        var result = await task;
        if (result)
        {
CrossPermissions.Current.OpenAppSettings();
        }

        return false;
    }

    request = true;
}

if (request || permissionStatus != PermissionStatus.Granted)
{
                                var newStatus = await
CrossPermissions.Current.RequestPermissionsAsync(Permission.Location);
                                if (newStatus.ContainsKey(Permission.Location)
&&newStatus[Permission.Location] != PermissionStatus.Granted)
        {
            var title = "Location Permission";
            var question = "To get your current city the location permission is
required.";
            var positive = "Settings";
            var negative = "Maybe Later";
            var task = Application.Current?.MainPage?.DisplayAlert(title,
question, positive, negative);
            if (task == null)
                return false;

```

```

        var result = await task;
        if (result)
        {
CrossPermissions.Current.OpenAppSettings();
        }
        return false;
    }
}

    return true;
}
}
}
public class LocalStorageConfiguration : IConfiguration
{
    private readonly ISyncLocalStorageService _localStorage;

    public LocalStorageConfiguration(ISyncLocalStorageService localStorage)
    {
        _localStorage = localStorage;
    }

    public string this[string key] {
        get => _localStorage.GetItem<string>(key);
        set => throw new NotImplementedException();
    }

    public IEnumerable<IConfigurationSection>GetChildren()
    {
        throw new NotImplementedException();
    }

    public IChangeTokenGetReloadToken()
    {
        throw new NotImplementedException();
    }

    public IConfigurationSectionGetSection(string key)
    {
        throw new NotImplementedException();
    }
}
}
public static class TestDataProvider
{

```

```

        public static void InitTestDataAsync (Android.App.Activity context)
        {
            var assetList = context.Assets.List ("").ToList ();
public static void InitTestDataAsync ()
{
    try
    {
        var locations = TestData.Locations;

                                var existingLocations =
SettingsStudio.Settings.LocationsJson?.GetLocations ();

        if (existingLocations?.Any () ?? false)
        {
            locations = existingLocations;
        }

        var selected = locations?.FirstOrDefault (l =>l.Selected);

        foreach (var location in locations)
        {
            var name = location.name.Split(',') [0].Replace (' ', '_');

            var path = Foundation.NSBundle.MainBundle.PathForResource
(name, "json");

                                if (!string.IsNullOrEmpty(path))
                                {
                                    using (var data =
Foundation.NSData.FromFile (path))
                                    {
                                        var json =
Foundation.NSString.FromData (data,
Foundation.NSStringEncoding.ASCIIStringEncoding).ToString ();
                                        var path = $"{name}.json";

                                if (assetList.Contains(path))
                                {
                                    using (var sr = new
System.IO.StreamReader (context.Assets.Open (path)))
                                    {
                                        var json = sr.ReadToEnd ();
                                        var weather =
json?.FromJson<WuWeather> ();

```

```

        if (weather != null)
        {
            WuClient.Shared.AddLocation
(new WuLocation (location, weather), true);
        }
    }
}

```

```

        var lastSelected = WuClient.Shared.Locations.FirstOrDefault (l
=>l.Location.name.CompareIgnoreCase (selected?.name) == 0);

```

```

        if (lastSelected != null)
        {
            WuClient.Shared.Selected = lastSelected;
        }
        else
        {
            var i = new Random ().Next (5);

```

```

            WuClient.Shared.Selected = WuClient.Shared.Locations [i];
        }
    }
    catch (Exception ex)
    {
        System.Diagnostics.Debug.WriteLine ("{"ex.Message}");
    }
}
}

```

```

public class LocalStorageConfiguration : IConfiguration
{
    private readonly ISyncLocalStorageService _localStorage;

    public LocalStorageConfiguration(ISyncLocalStorageService localStorage)
    {
        _localStorage = localStorage;
    }

    public string this[string key] {
        get => _localStorage.GetItem<string>(key);
        set => throw new NotImplementedException();
    }
}

```

```

public IEnumerable<IConfigurationSection>GetChildren()
{
    throw new NotImplementedException();
}

public IChangeTokenGetReloadToken()
{
    throw new NotImplementedException();
}

public IConfigurationSectionGetSection(string key)
{
    throw new NotImplementedException();
}
}
}

public class PinnedLocationsService
{
    private const string PinnedLocationsKey = "PINNED_LOCATIONS";
    private readonly ILocalStorageService _localStorage;

    public PinnedLocationsService(ILocalStorageService localStorage)
    {
        _localStorage = localStorage;
    }

    public async Task<IList<Location>>LoadPinnedLocations()
    {
        if (await _localStorage.ContainKeyAsync(PinnedLocationsKey))
        {
            var pinnedLocations = await
            _localStorage.GetItemAsync<Location[]>(PinnedLocationsKey);
            return new List<Location>(pinnedLocations);
        }
        else
        {
            return new List<Location>();
        }
    }

    public Task SavePinnedLocations(IEnumerable<Location>pinnedLocations)
    {
        return _localStorage.SetItemAsync(PinnedLocationsKey,
        pinnedLocations);
    }
}

```

}  
}