

Міністерство освіти і науки України  
Національний технічний університет  
«Дніпровська політехніка»  
Навчально-науковий інститут електроенергетики  
(інститут)  
Електротехнічний факультет  
(факультет)  
Кафедра кіберфізичних та інформаційно-вимірювальних систем  
(повна назва)

**ПОЯСНЮВАЛЬНА ЗАПИСКА**  
**кваліфікаційної роботи ступеню магістра**

студента Чорномора Ростислава Руслановича

(піб)

академічної групи 151М-21-1

(шифр)

спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології

(код і назва спеціальності)

за освітньо-професійною програмою 151 Автоматизація та комп'ютерно-інтегровані технології

(офіційна назва)

на тему Автоматизація процесів керування сонячних станцій на основі керуючих моделей

(назва за наказом ректора)

Консультанти	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
Керівник кваліфікаційної роботи	проф. Ткачов В.В.			
розділів:				
Синтез системи керування	доц. Бубликов А.В.			
Експериментальний розділ	ст. викл. Бойко О. О.			
Економічна частина	ст. викл. Яремчук І.О.			
Охорона праці	проф. Чеберячко Ю.І.			

Рецензент				
-----------	--	--	--	--

Нормоконтролер	ас. Славінський Д.В.			
----------------	----------------------	--	--	--

**Дніпро**  
**2022**

**ЗАТВЕРДЖЕНО:**

Завідувач  
кафедри  
кіберфізичних та

інформаційно-вимірювальних систем

(повна назва)

\_\_\_\_\_ Бубліков А.В.

(підпис)

(прізвище, ініціали)

« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ року

**ЗАВДАННЯ**

**на кваліфікаційну роботу  
ступеня магістра**

студенту Чорномору Р.Р. академічної групи 151М-21-1  
(прізвище та ініціали) (шифр)

спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології

за освітньо-професійною програмою 151 Автоматизація та комп'ютерно-інтегровані технології

(офіційна назва)

на тему Автоматизація процесів керування сонячних станцій на основі керуючих моделей

затверджену наказом ректора НТУ «Дніпровська політехніка» від \_\_№ \_\_с

Розділ	Зміст	Термін виконання
Стан питання та постановка завдання	На основі матеріалів виробничих практик, інших науково-технічних джерел сформулювати наукове завдання, конкретизувати предмет та мету досліджень	01.10.20 – 07.10.20
Теоретичний	Обґрунтувати теоретичну базу розв'язання наукового завдання, якому присвячено роботу	07.10.20 – 15.10.20
Синтез системи керування	Провести синтез системи та її дослідження із використанням методів та інструментів, що обґрунтовані та створені у теоретичному розділі	15.10.20 – 01.11.20
Експериментальний	Розробка програмного забезпечення системи керування на підставі обраного регулятора та його налаштувань	01.11.20 – 15.11.20
Економічна частина	Обґрунтувати доцільність витрат на створення та впровадження системи керування	15.11.20 – 25.11.20
Охорона праці	Розробити організаційно-технічні заходи щодо реалізації правил безпеки при експлуатації системи	25.11.20 – 07.12.20

**Завдання видано**

\_\_\_\_\_

(підпис керівника)

проф. Ткачов В.В.

(прізвище, ініціали)

**Дата видачі**

**Дата подання до атестаційної комісії**

**Прийнято до виконання**

\_\_\_\_\_

(підпис студента)

Чорномор Р.Р.

(прізвище, ініціали)

## ПЛАН

Вступ

Розділ 1

1.1 Об'єкт дослідження

1.1.1 Відновлювані джерела електроенергії

1.1.2 Відновлювані джерела та їх місце в електроенергетиці, роль та динаміка розвитку

1.1.3 Принципи фотогальванічного перетворення енергії сонця та основне обладнання для їх здійсненн

1.1.4 Основні завдання системи моніторингу та диспетчеризації сонячної електростанції

1.2 Предмет дослідження

1.2.1 Специфічні особливості ринку електроенергії у зв'язку з проблемою її зберігання та складування

1.3 Мета та завдання дослідження

1.4 Актуальність

Розділ 2

2.1 Базові методи для прогнозування генерації електроенергії під час роботи сонячної електростанції

2.1.1 Методи прогнозування сонячної активності терміном до 6 годин

2.1.2 Метод Total sky imagery

2.1.3 Метод аналізу хмарної ситуації за знімками з космосу

2.1.4 Прогнозування сонячної активності терміном до 1 доби

2.1.5 Просторово-часова інтерполяція та згладжування

2.1.6 Погодинний прогноз

2.1.7 Види тимчасових рядів

2.1.8 Регресійні моделі для виділення тренду

2.1.9 Модель ARIMA

2.1.10 Тета-модель

2.1.11 Використання штучних нейронних мереж

2.2 Моделі прозорого, чорного та сірого ящика

2.2.1 Тестування білої скриньки

2.2.2 Тестування чорного ящика

2.2.3 Методи тестування програми як "білої скриньки" та "чорної скриньки

2.2.4 Тестування сірої скриньки

2.3 Алгоритми ансамблевого прогнозування

2.4 Основна ідея дослідження

Розділ 3

3.1 Концепція побудування математичної моделі процесу генерації сонячної енергії.

3.2 Алгоритм обчислення рівня сонячної інсоляції.

3.3 Розробка машинно-навченої прогностичної моделі процесу генерації фотоелектричної енергії з урахуванням метеорологічних факторів, що випадково змінюються.

Розділ 4

4.1 Принцип роботи

4.1.1 Мережеві системи моніторингу

4.1.2 Автономні системи сонячного моніторингу

4.1.3 Перевірка продуктивності

4.1.4 Оцінка системи

4.2 Функціональні можливості АСКОЕ

4.3 Інвертор Струму SUNNY TRIPOWER 15000TL/20000TL/ 25000TL

4.3.1 Опис виробу

4.3.2 Інтерфейси та функції

4.3.3 Технічні характеристики

4.4 Контролер MCL 5.8 із внутрішнім 4-частотним GSM/GPRS

модемом

4.4.1 Функціональне призначення та застосування контролера MCL 5.8

4.4.2 Основні модулі контролера

4.4.3 Основні характеристики інтегрованого GSM/GPRS модуля

4.4.4 Основні модулі контролера

4.4.5 Технічні характеристики контролера MCL 5.8

4.5 Багатофункціональний лічильник активної, реактивної та повної енергії  
EPQS

4.5.1 Призначення та особливості лічильника

4.5.2 Функціонал лічильника EPQS

Розділ 5

5.1 Розробка програмного забезпечення

6. Економічне обґрунтування

6.1 Визначення трудомісткості обробки програмного продукту

6.2 Розрахунок витрат за створення програмного продукту

6.3 Висновки у розділі

## 7. Охорона праці та захист населення в надзвичайних ситуаціях

### 7.1 Основи охорони праці

### 7.2 Захист населення у надзвичайних ситуаціях

Перелік використаних джерел

Загальні висновки

#### Додаток А

##### А.1 Програмне забезпечення об'єкта

###### А.1.1 Список програм

###### А.1.2 Основна програма

###### А.1.3 Квадратичний прогноз $Q(t)$

###### А.1.4 Довірчий прогноз $T(t)$

###### А.1.5 Рефлексивний прогноз $R(t)$

###### А.1.6 Порівняння прогнозів та динамічна корекція прогнозу

#### Додаток Б

Алгоритм роботи програми



## ВСТУП

Безперервне зростання цін на традиційні енергоносії і на електричну енергію, що отримується в основному від спалювання викопного палива, обумовлений насамперед зростанням собівартості палива, що видобувається, і збільшенням витрат на його транспортування. У той самий час намітилася стала тенденція зниження вартості енергії, одержуваної від поновлюваних джерел. Одним із видів альтернативної енергетики є сонячна енергія, яка набула величезної популярності, адже вона є джерелом децентралізованого енергопостачання, що дозволяє їй виробляти електрику для об'єктів, віддалених від загальної енергосистеми.

Якісні прогнози майбутнього вироблення електрики дозволяють виробникам електроенергії та мережним операторам активно керувати змінною продуктивністю сонячних електростанцій, тим самим оптимально інтегруючи сонячні ресурси до загальної енергосистеми країни.

Прогноз обсягів вироблення електроенергії сонячною станцією – це передусім прогнозування кількості сонячної радіації, яку отримують сонячні панелі. Вона залежить від багатьох факторів, основними з яких є кліматичні та метеорологічні умови – положення сонця на небі, тривалість світлового дня, хмарність, опади, сила вітру тощо.

Вибір методу прогнозу також залежить від цього, які саме дані необхідно отримати у результаті. Наприклад, прогноз сумарного вироблення електроенергії за певну кількість часу (за годину, кілька годин або добу) або зміна обсягів генерації протягом певного відрізка часу вимагають використання різних математичних моделей та підходів. Причини створення програми прогнозування

Про встановлення «зелених» тарифів на електричну енергію, вироблену генеруючими установками приватних домогосподарств

Спеціальні умови, за якими держава купує у власників домашніх СЕС надлишки електроенергії, називаються зеленим тарифом. Його величина залежить від періоду підключення до СЕС і поступово знижується. Але як би там не було, цей тариф залишається досить високим, щоб забезпечити швидку окупність сонячних батарей в Україні.

З 1го січня 2020 року він становить 0,16 Євро (при курсі 33грн/євро — 5,28 грн) за 1 кВт\*год, а з початку 2025 до 31 грудня 2029 року становитиме 0,14 Євро (при курсі 33грн./ євро - 4,62 грн / кВт \* год). Потім програма може бути пролонгована або скасована. Надалі держава також купуватиме у виробників надлишки енергії, можливо, не за таким високим тарифом.

Згідно з постановою Національної комісії, що здійснює державне регулювання у сферах енергетики та комунальних послуг, від 30 серпня 2019 року № 1817, Національна комісія постановила:

Установити «зелений» тариф на електричну енергію, вироблену з енергії сонячного випромінювання генеруючими установками приватних домогосподарств, встановлена потужність яких не перевищує 30 кВт та які введені в експлуатацію:

з 01 квітня 2013 року по 31 грудня 2014 року - 1303,52 коп/кВт·год (без ПДВ);

з 01 січня 2015 року по 30 червня 2015 року - 1172,38 коп/кВт·год (без ПДВ);

з 01 липня 2015 року по 31 грудня 2015 року - 728,09 коп/кВт·год (без ПДВ);

з 01 січня 2016 року по 31 грудня 2016 року - 690,90 коп/кВт·год (без ПДВ);

з 01 січня 2017 року по 31 грудня 2019 року - 657,63 коп/кВт·год (без ПДВ);

з 01 січня 2020 року по 31 грудня 2024 року - 591,08 коп/кВт·год (без ПДВ).

Для порівняння приведу дані цін на електроенергію для побутових споживачів

Згідно до постанови кабінету Міністрів України № 859 від 11.08.2021 встановлено для побутових споживачів в Україні з 01.10.2021 нові ціни на електроенергію. Наразі домогосподарства, що споживають на місяць до 250 кВт·год (включно), платитимуть за електроенергію за тарифом 1,44 грн. (з ПДВ) за 1 кВт·год, а ті, що споживають понад 250 кВт·год — за тарифом 1,68 грн. (з ПДВ) за 1 кВт·год. Термін дії цих тарифів було встановлено до 30.04.2022, але постановою № 453 від 16.04.2022 його подовжено до 31 жовтня 2022 року.

Рис.1 Динаміка росту кількості сонячних установок

Якщо звернути увагу на приведений вище графік, то можна помітити експоненційний ріст кількості СЕС(сонячних електро-станцій)



Дані були взяті з відкритих джерел, станом на 15.11.2018

Потужність станції: 30,25 кВт

Власне споживання: 200 кВт\*ч/місяць

Вартість електроенергії: 1,68 грн./кВт\*год

Величина “зеленого тарифу”: 5,72 грн./кВт\*год

Податкова ставка: 19,5%

Підвищення ціни електроенергії: 15%/рік

Відсоток деградації сонячної панелі в рік: 0,7%/рік

Курс долару: 28,1 грн.

Початкові інвестиції: 24 541\$

Финансовые показатели станции по годам					
Год	Доход от продажи [\$]	Налог [\$]	Прибыль от продажи [\$]	Экономия от собств. потребл. [\$]	Финансовый результат [\$]
1-й	6 055	1 181	4 874	144	-19 523
2-й	6 009	1 172	4 837	166	-14 520
3-й	5 964	1 163	4 801	190	-9 529
4-й	5 918	1 154	4 764	219	-4 546
5-й	5 873	1 145	4 728	252	434
6-й	5 828	1 136	4 692	290	5 416
7-й	5 784	1 128	4 656	333	10 405
8-й	5 740	1 119	4 621	383	15 409
9-й	5 697	1 111	4 586	440	20 435
10-й	5 653	1 102	4 551	507	25 493
11-й	5 610	1 094	4 516	583	30 592
12-й	5 567	1 086	4 481	670	35 743
13-й	5 525	1 077	4 448	770	40 961

Рис.2 Фінансові показники станцій по рокам

Враховуючи початкові інвестиції та податки, котрі будуть перекивати прибутки за перші 4 роки, приватному підприємцю знадобиться більш ніж чотири роки до першого отриманого прибутку за допомогою “зеленого тарифу”

Розглянемо прогнозування обсягів відпуску електричної енергії виробниками за «зеленим» тарифом згідно до постанови 13.12.2019 № 2804

1. Споживачі за «зеленим» тарифом до 9:00 за день до торгового дня надають гарантованому покупцю погодинні добові графіки відпуску електричної енергії (прогнозні обсяги) та доступну потужність генеруючих

одиниць з розбивкою по технологіях, тарифах/видах генерації/за видом альтернативного джерела та по географічних регіонах.

2. У випадку технічного обслуговування програмного забезпечення ОР (далі - система) споживачі за «зеленим» тарифом для тих розрахункових періодів, для яких на час подання заявок система буде недоступною, надають гарантованому покупцю оновлені графіки відпуску електричної енергії (прогнозні обсяги) та доступну потужність генеруючих одиниць з розбивкою по технологіях, тарифах/видах генерації/за видом альтернативного джерела та по географічних регіонах не пізніше ніж за три години 45 хвилин до часу початку технічного обслуговування системи.

3. Споживачі за «зеленим» тарифом починаючи з 15:00 дня, що передує торговому, але не пізніше ніж за дві години 45 хвилин до розрахункового періоду надають гарантованому покупцю оновлений графік відпуску електричної енергії та інформацію про доступну потужність генеруючих одиниць з розбивкою по технологіях, тарифах/видах генерації/за видом альтернативного джерела та по географічних регіонах для кожного розрахункового періоду торгового дня.

4. Прогнозні обсяги, що надаються споживачами за «зеленим» тарифом гарантованому покупцю, повинні відповідати обсягам, що зазначаються у графіках фізичного відпуску електричної енергії, що надаються споживачами за «зеленим» тарифом ОСП.

5. Порядок надання споживачами за «зеленим» тарифом даних щодо прогнозних обсягів відпуску визначається гарантованим покупцем в інструкції з надання прогнозних даних, що публікується на його вебсторінці.

6. Прогнозні обсяги відпуску, що подані споживачем за «зеленим» тарифом гарантованому покупцю, повинні відповідати фізичним можливостям генеруючого обладнання. За наявності невідповідності такі прогнозні обсяги не враховуються гарантованим покупцем при подачі заявок на ринку електричної енергії, а у разі відсутності попередньо наданих споживачем за

«зеленим» тарифом прогнозних обсягів відпуску вони визначаються на рівні нульового відпуску.

Платіж за електричну енергію споживачу за «зеленим» тарифом за розрахунковий місяць визначається за формулою

$$P_m = \sum_p \sum_d^m \sum_h^d (W_h^u \cdot \frac{T_m^u}{100}), \text{ грн,}$$

p	споживач за «зеленим» тарифом;
u	генеруюча одиниця;
m	розрахунковий місяць;
d	доба розрахункового місяця;
h	година доби розрахункового місяця;
$W_h^u$	обсяг електричної енергії, відпущеної за «зеленим» тарифом за розрахунковий місяць, кВт·год;
$T_m^u$	«зелений» тариф, встановлений для генеруючої одиниці за «зеленим» тарифом, коп/кВт·год.

Виходячи з приведеної вище інформації можна прийти до висновку що приватні підприємці залежні напряму в своєму прогнозуванні прибутків від прогнозування продуктивності СЕС, отже потребують якісної, точної та адаптивної системи прогнозування обсягів вироблення електроенергії в короткостроковій та довгостроковій перспективах. Метою дипломного проекту є розробка комп'ютерної системи моніторингу та системи прогнозування процесів вироблення сонячної електроенергії, що навчається.

## Розділ 1

### 1.1 Об'єкт дослідження

#### 1.1.1 Відновлювані джерела електроенергії.

Термін відновлювані джерела енергії (ВДЕ) застосовується по відношенню до тих джерел енергії, запаси яких заповнюються природним чином, насамперед,

за рахунок потоку енергії сонячного випромінювання, що надходить на поверхню Землі, і в найближчій перспективі є практично невичерпними. Це насамперед сама сонячна енергія, а також її похідні: енергія вітру, енергія рослинної біомаси, енергія водних потоків тощо. До відновлюваних джерел енергії відносять також геотермальне тепло, що надходить на поверхню Землі з її надр, низькопотенційне тепло навколишнього середовища, яке можна використовувати, наприклад, за допомогою теплових насосів, а також деякі джерела енергії, пов'язані з життєдіяльністю людини (теплові «відходи» житла). органічні відходи промислових та сільськогосподарських виробництв, побутові відходи тощо).

### **1.1.2 Відновлювані джерела та їх місце в електроенергетиці, роль та динаміка розвитку**

За підсумками 10 місяців 2020р. встановлена потужність ВЕС та СЕС становить 1,11 ГВт та 5,98 ГВт, а фактичне виробництво 2,57 млрд.кВт год та 5,78 млрд.кВт год відповідно.

Загалом сектор альтернативної енергетики України налічує 881 суб'єкт господарювання, для яких встановлено "зелені" на електричну енергію, вироблену на об'єктах електроенергетики.

Починаючи з 1 квітня 2020р. «зелені» тарифи на електричну енергію, що виробляється суб'єктами господарювання на об'єктах електроенергетики, які використовують альтернативні джерела енергії, та надбавки до «зелених» тарифів за дотримання рівня використання обладнання українського виробництва, встановлені НКРЕКП на рівні від 2,1954 грн/кВт•год (з ПДВ) до 15,807 грн/кВт•год (з ПДВ), тобто середній «зелений» тариф становить 9,0012 грн/кВт•год (з ПДВ). Водночас, вартість «зеленої» електричної енергії в країнах ЄС становить близько 4,15 грн/кВт•год (з усіма податками).

Сумарний обсяг інвестицій міжнародних компаній [20] оцінюється на рівні €2,5 млрд. – це один із найкращих, за період незалежності України, прикладів успішного залучення іноземних інвестицій у національну економіку. При цьому

за три квартали 2020р. до сектору альтернативної енергетики надійшло інвестицій на суму \$1,2 млрд.

За результатами 2019р., учасниками ринку альтернативної енергетики до бюджетів усіх рівнів було сплачено 19,8 млрд.грн, що втричі більше від сплати податків вугільним сектором. У період 2009-2019рр, операторами установок ВДЕ до державного бюджету було сплачено 93,6 млрд.грн, що на 45,4 млрд.грн більше за рівень сплати податків вугледобувними та теплогенеруючими підприємствами.

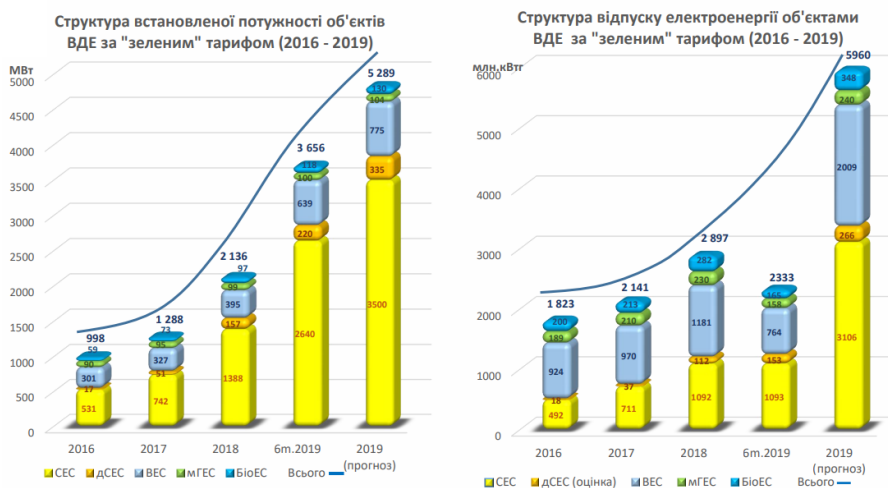


Рис.1.1. Діаграма встановленої потужності об'єктів та виробництва енергії цими об'єктами



### 1.1.3 Принципи фотогальванічного перетворення енергії сонця та основне обладнання для їх здійснення

Сонячна батарея (батарея сонячних елементів) - пристрій, що безпосередньо перетворює енергію сонячного випромінювання на електричну. Дія сонячного елемента (СЕ) ґрунтується на використанні явища внутрішнього фотоефекту. Найбільше застосування отримали конструкції СЕ з р-п-переходами та гетеропереходами, що є базовою напівпровідниковою пластиною з тонким фронтальним шаром напівпровідника, що має тип провідності, протилежний типу провідності базової області. При опроміненні в напівпровіднику генеруються доповн. носії заряду, що переміщуються під дією електричного поля р-п - переходу та створюють на зовнішніх висновках фотоедс.

Найважливішим параметром СЕ є його ефективність перетворення енергії сонячного випромінювання на електричну

$\eta = (I_n * U_n) / P_c$ , де  $P_c$  - потужність сонячного випромінювання, що падає на поверхню СЕ. Ефективність СЕ визначається тим, що частина сонячного випромінювання з енергією фотона, меншої ширини забороненої зони напівпровідника, проходить через СЕ без поглинання і фотоелектрич. Відношення є марною. Чим менша ширина забороненої зони, тим більша частка сонячного світла поглинається в ньому.

Мережеві інвертори перетворюють електроенергію постійного струму на енергію змінного струму, придатну для подачі в загальну електромережу. Мережевий інвертор повинен стежити за фазою мережі і, з дуже високою точністю, безперервно підтримувати вихідну напругу трохи вище напруги мережі. Високоякісний сучасний мережевий інвертор має фіксований коефіцієнт потужності - він видає прецизійну вихідну напругу та струм, а випередження фази знаходиться в межах 1 градуса від мережі змінного струму.

Інвертор управляється мікропроцесором, який стежить за поточною формою напруги змінного струму і виводить напругу, що точно відповідає напругі мережі. Також необхідна подача реактивної потужності в мережу для підтримки напруги локальної мережі в межах допустимих значень. Інакше можливі перенапруги в мережі потужних систем, коли генерація енергії досягає свого піку, наприклад, близько полудня від сонячних панелей.

Існує кілька типів інверторів:

1. Мережеві (on-grid), які можуть працювати синхронізовано із централізованою мережею електропостачання. У разі відсутності живлення від мережі робота інвертора припиняється. Даний тип інверторів дозволяє системі працювати без акумуляторних батарей та продавати електроенергію за «зеленим тарифом». Автономные (off-grid) инверторы не работают с центральной электросетью и предназначены для преобразования электрического тока в автономных солнечных электростанциях с аккумуляторами.
2. Гібридні (hybrid) або «акумуляторно-мережеві» перетворювачі, що поєднує функції автономних і мережевих інверторів. Цей тип інверторів дозволяє підключати систему до загальної мережі, продавати електроенергію за «зеленим тарифом», а також працювати паралельно з нею, виконуючи роль резервного живлення, використовуючи акумуляторні батареї.

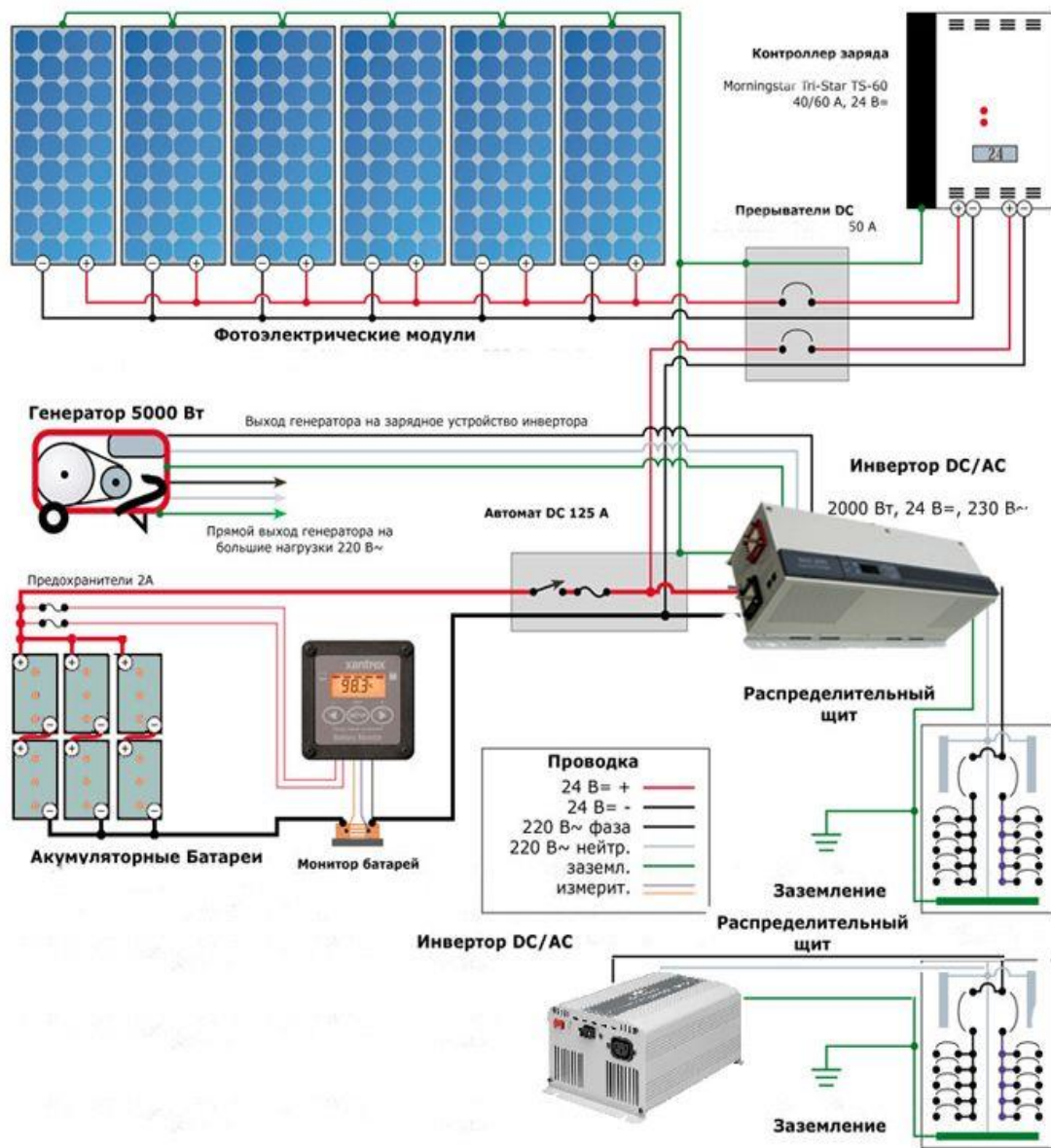


рис.1.2

### **1.1.4 Основні завдання системи моніторингу та диспетчеризації сонячної електростанції:**

- Інформація про продуктивність сонячної електростанції
- Бізнес-аналітика – порівняння ключових індикаторів продуктивності різних ділянок сонячної станції/ різних сонячних станцій
- Контролює поточні параметри роботи сонячної електростанції
- Експорт даних у треті системи
- Звітність про вироблену електроенергію за періодами з побудовою графіків
- Контроль стану обладнання сонячної інверторної підстанції
- Дистанційний моніторинг та управління
- Архівування даних та дій оператора
- Контроль роботи всього обладнання сонячної електростанції
- Моніторинг виробничого циклу електроенергії кожного струнгу/кожної інверторної підстанції
- Діагностика силового обладнання
- Отримання та обробка сигналів тривоги та повідомлень
- Дистанційний моніторинг та управління

Значну роль ринкових умов займає автоматизована система комерційного обліку електроенергії (АСКУЭ), що також є підсистемою АС СО. АСКОЕ призначена для здійснення моніторингу та комерційного обліку балансів потужності та енергії суб'єктів ринку та забезпечує отримання даних про середні значення потужності та про враховану електроенергію за зонами доби за кожен календарну добу та накопичувально за тиждень, місяць, рік. Прилади комерційного обліку встановлюються на енергооб'єктах та забезпечує передачу накопиченої інформації каналами зв'язку на найближчий за ієрархією диспетчерський пункт. Інформація надходить у комп'ютер, обробляється (агрегується) і передається наступний рівень управління, до верхнього, соціальної та інформаційно-обчислювальної систему енергозбуту.

Система PV SCADA є важливою частиною фотоелектричної сонячної електростанції. Система PV SCADA виконує всі функції збору даних, моніторингу та керування електростанцією.

Типова структурна схема PV SCADA показано на рис. 3

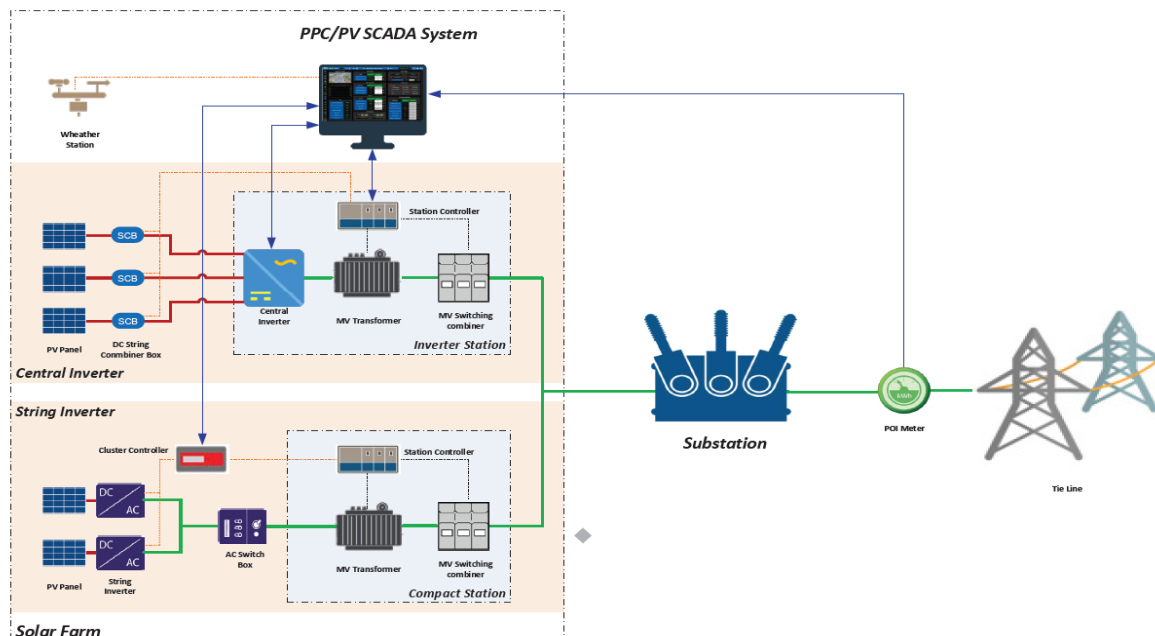


Рис. 1.3

На кожній фотоелектричній станції встановлюються:

- \* Контролер станції з аналоговим, цифровим входом/виходом та функцією перетворення протоколу підтримки для збору всіх даних моніторингу та управління фотоелектричною електростанцією з моніторів блоку об'єднання постійного струму, інверторів, трансформатора середньої потужності, панелі блоку кільцевої магістралі, реле захисту, багатофункціональних лічильників, метеостанції.
- \* Метеорологічні станції для отримання метеорологічної інформації для оцінки продуктивності та прогнозування вироблення електроенергії на фотоелектричній електростанції. Ці метеостанції будуть підключені до пристроїв збирання даних на інверторних станціях.

У диспетчерській операторі встановлюються:

- \* Сервери керування електростанцією та SCADA для збору даних, обробки даних, зберігання історичних даних, моніторингу та керування всією фотоелектричною електростанцією.
- \* Сервер бази даних історії для зберігання даних та історичних даних
- \* Годинник із супутниковою синхронізацією для синхронізації часу всього обладнання в системі PV SCADA.

## **1.2 Предмет дослідження.**

### **1.2.1 Специфічні особливості ринку електроенергії у зв'язку з проблемою її зберігання та складування.**

Найбільш важливими особливостями економіки енергосистем, викликаними специфікою електроенергії як товару та які необхідно враховувати при організації ринку електроенергії, є:

- 1) передача, розподіл та споживання електроенергії в силу її фізичної природи відбуваються практично одночасно і її неможливо складувати у значних обсягах.
- 2) електроенергія є найвищою мірою стандартизованим продуктом, що поставляється безліччю виробників у загальні електричні мережі і миттєво споживається звідти ж безліччю споживачів. Тому з фізичної точки зору неможливо визначити, хто виробив електроенергію, що споживається тим чи іншим споживачем — можна лише контролювати обсяги постачання у загальну мережу від кожного виробника та обсяги споживання з неї кожним споживачем;
- 3) електроенергія, одержувана споживачем з енергосистеми, є товаром першої необхідності, тільки в окремих випадках мають інші товари-замінники (наприклад, перехід на електропостачання від автономної дизельної електростанції, переведення електроопалення на газове опалення та деякі інші випадки).
- 4) виробники виробляють і поставляють у загальну мережу електричну потужність відповідно до своїх зобов'язань, а всі споживачі сумарно споживають електричну потужність відповідно до своїх зобов'язань (або прогнозом диспетчера). Але на практиці в

силу різного роду обставин, як виробники, так і споживачі допускають відхилення від своїх зобов'язань.

Це спричиняє дисбаланс між постачанням та споживанням, що може призводити до втрати стійкості ринку. У зв'язку з цим одним із найважливіших завдань моніторингу електроенергетичних процесів є завдання прогнозування вироблення та споживання електричної енергії.

Специфіка електроенергії як товару призводить до розвитку ринку електроенергії, відмінного від звичайних товарних ринків.

Предметом дослідження є проблема побудови комп'ютерної системи моніторингу, що самонавчається, і прогнозування процесів вироблення сонячної електроенергії, що забезпечує мінімізацію помилок прогнозу.

### **1.3 Мета та завдання дослідження**

Метою роботи є розробка комп'ютерної системи моніторингу та прогнозування процесів вироблення сонячної електроенергії, що відповідає вимогам енергоринку щодо точності та своєчасності прогнозів.

У роботі ставляться і вирішуються такі завдання.

1. Розробка та дослідження прогностичної самонавчається математичної моделі фотогальванічної станції.
2. Дослідження імітаційної моделі теплофізичних процесів, які впливають ефективність PV перетворення сонячними панелями.
3. Розробка структурної схеми та вибір обладнання для системи моніторингу сонячної електростанції потужністю .....мВт
4. Розробка програмного забезпечення системи.
5. Аналіз результатів експериментального моніторингу та прогнозу вироблення електроенергії на діючій фотогальванічній станції.



## 1.4 Актуальність

Зростання попиту електроенергію має стійку кореляцію зі зростанням ВВП і зростанням промислового виробництва. Відношення приросту електроспоживання до приросту ВВП називається еластичність електроспоживання і має відносно постійну величину для тривалих періодів.

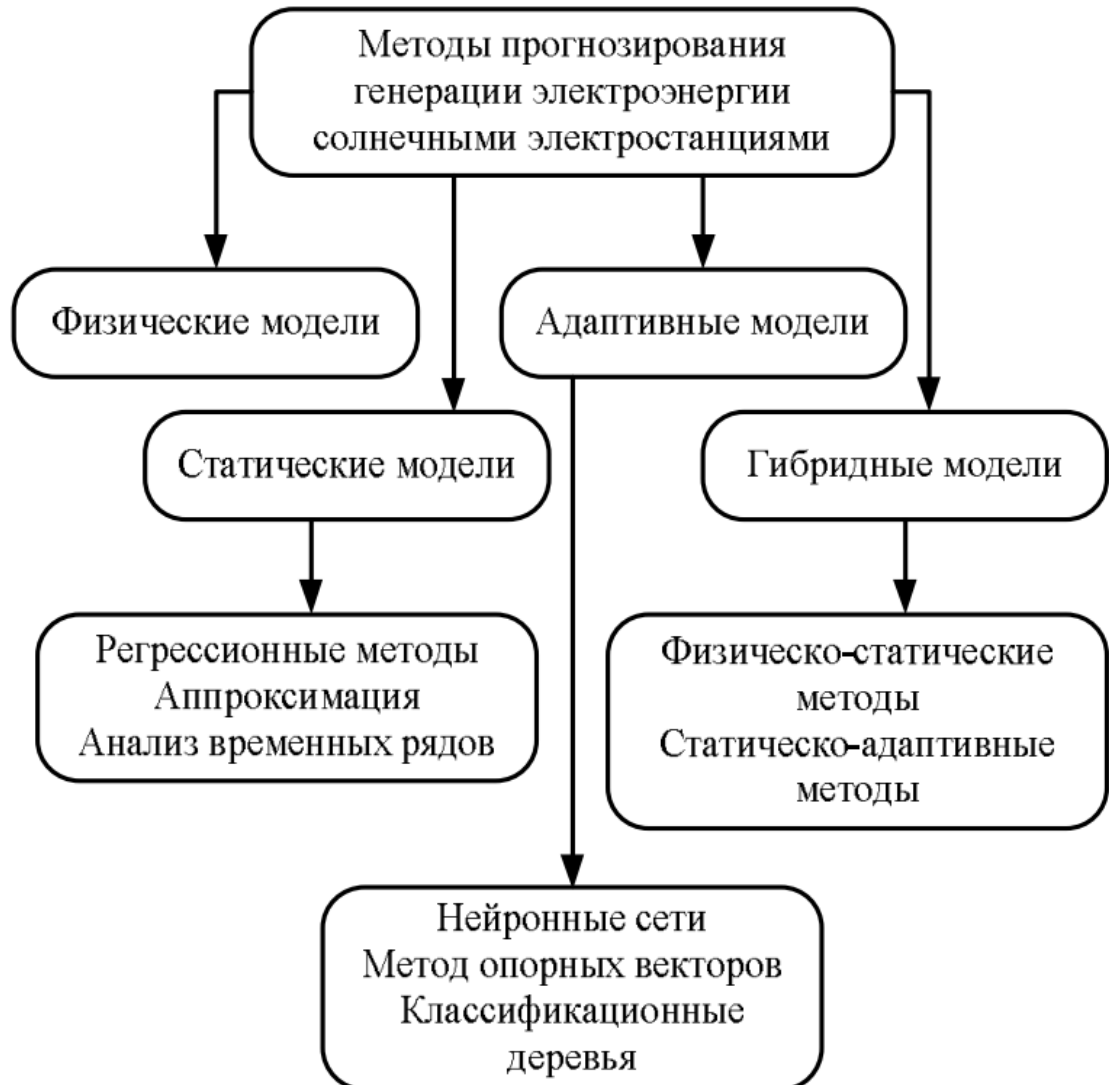
У 2018 році загальна потужність сонячних фотоелектричних установок сягнула 1545 МВт. Найбільше зростання відбулося у 2019 році, коли загальна встановлена потужність збільшилася у 3,5 рази порівняно з 2018 роком. Фотоелектричні системи загального призначення склали левову частку цього розгортання, причому 3,53 ГВт було встановлено в річному обчисленні. У середині 2020 року уряд запровадив ретроактивні обов'язкові скорочення для всіх сонячних електростанцій. Більше того, власники систем зіткнулися з нестабільною окупністю за схемою FIT. В даний час 35% все ще не виплачено за 2020 рік, а заборгованість за 2021 становить близько 15%. В результаті розробка нових РЕ потужності значно скоротилися у 2020 та 2021 роках. Минулого року український сонячний сектор додав 1169 МВт нежитлової сонячної фотоелектричної потужності та 226 МВт житлової фотоелектричної потужності, загалом 1395 МВт нововстановленої сонячної енергії у 2020 році (див. рис. ГВ15.1). Це на 60% менше, ніж торік. У першому кварталі 2021 року було додано лише 105 МВт більших фотоелектричних систем. У сукупному вираженні 6873 МВт сонячної енергії було встановлено в Україні до 1 січня 2021 року, з 779 МВт або 11 % цієї потужності, що складається з житлових фотоелектричних установок (не більше 30 кВт у розмірі)<sup>1</sup>. Домашні фотоелектричні установки продовжували своє зростання в 2020 році, але приріст приблизно на 0,2 ГВт становить лише близько половини

встановленої потужності у попередньому році. За оцінками українського уряду, до кінця 2020 року в країні було встановлено 30 тисяч домашніх систем. У всіх нових технологіях ВІЕ в Україні, за винятком великих гідроелектростанцій, більшість електроенергії виробляється із сонячних фотоелектричних батарей. У період з січня 2021 року по червень 2021 року сонячні фотоелектричні системи виробили 926,8 ГВт-год зеленої електроенергії.

Енергетика є галуззю економіки, що є індикатором рівня розвитку, науки і держави загалом. Людство за всю історію свого існування витратило приблизно 950 трлн кВт/год енергії всіх видів, 2/3 витраченої енергії припадає на останні 30 років. Тому проблема освоєння нетрадиційних та відновлюваних джерел енергії стає дедалі актуальнішою. Розробка комп'ютерної системи моніторингу та прогнозування процесів вироблення сонячної електроенергії є необхідною для підтримки конкурентоспроможності, підвищення продуктивності українських компаній сонячної енергетики, її реалізації на ринку.

## Розділ 2

### Базові методи для прогнозування генерації електроенергії під час роботи сонячної електростанції



Класифікація методів прогнозування генерації електроенергії СЕС

1. Фізичні моделі описують фізичні між погодними умовами та сонячною радіацією, отриманими з використанням чисельного прогнозу погоди та виробленням електричної енергії на станції.
2. Статистичні моделі описують взаємозв'язок між щільністю потоку сонячного випромінювання, отриманим з використанням чисельного прогнозу погоди, та генерацією електричної енергії на сонячній електростанції путем статистичного

аналізу часових рядів ретроспективних даних без урахування фізичних факторів

3. Адаптивні моделі використовують системи штучного інтелекту з метою визначення зв'язку між прогнозованими погодними умовами та вихідною потужністю електростанції.
4. Гібридні моделі найчастіше представляють комбінацію фізичних і статистичних моделей.

### **2.1.1 Методи прогнозування сонячної активності терміном до 6 годин**

Розглянемо два основних методи прогнозування сонячної активності, які використовуються для короткострокових (до 6 годин) прогнозів та подальшого передбачення майбутньої генерації від сонячної електростанції на строк менш як добу.

### **2.1.2 Метод Total sky imagery**

Total sky imagery – метод, який використовується для прогнозування генерації електроенергії сонячною електростанцією буквально в режимі реального часу. Цей метод дозволяє досить точно передбачити виробіток на 10-30 хвилин вперед. Прогноз вироблення електроенергії на основі аналізу зображення піднебіння складається з 4 операцій:

- Отримання зображення піднебіння в районі встановлення сонячної електростанції – проводиться фотографування хмарної ситуації з поверхні землі;
- Аналіз отриманих даних, визначення типу хмар (розрізняють тонкі та товсті хмари);
- Оцінка вектора руху хмар – для цього використовується послідовність кількох знімків; використовуючи розташування хмар, а також отримані дані про вектор руху для короткострокового імовірного хмарного покриття,

розраховує потужність сонячного опромінення і створюється прогноз генерації електрики.

Спеціалізовані метеорологічні служби типу Whole Sky Imager разом із фотографіями розташування хмар надають докладну інформацію про масштаби, структуру та рух хмар. Маючи це дані, можна скласти дуже точний прогноз, але у досить маленькому тимчасовому горизонті – до 30 хвилин. Точність прогнозу більший час різко знижується через те, що дуже важко передбачити розвиток рух хмарного поля, і навіть зміна геометрії хмар. На сьогоднішній день немає чіткої математичної моделі розвитку хмарних полів. Як варіант – можна використовувати зображення з хмарною обстановкою, яка сфотографована у прилеглих областях та провести прогноз із урахуванням вектора переміщення хмар. Але при цьому слід врахувати, що хмари різних шарів мають різні характеристики, наприклад, низькі хмари пересуваються значно швидше, а ось високі - значно повільніше. Ще одна складність - на фото нижні хмари можуть закривати високі - все це в цілому позначається на точності прогнозу.

### **2.1.3 Метод аналізу хмарної ситуації за знімками з космосу**

Загалом ідея другого методу повторює Total sky imagery, лише замість фотографій неба та хмарної обстановки, отриманих із землі, використовують зображення, отримані з космічних супутників. При цьому зображення хмар може бути отримано як у традиційному вигляді (оптична зйомка), так і з використанням інфрачервоних датчиків. Основна перевага цього методу – отримання масштабів хмарності на значно більшій площі, до того ж високоякісні супутникові зйомки доступніші і охоплюють практично всю територію планети, тоді як Total sky imagery реалізовано на обмеженій території. Крім того, завдяки точному виміру коефіцієнта відображення, отриманого із супутника, можна дуже точно розрахувати індекс хмари, який пропорційний

оптична глибина (прозорість) хмари. Цей метод досить добре вивчений і активно застосовується щодо та нанесення на карту сонячних ресурсів – визначення інсоляції для місцевості.

До недоліків цього можна віднести той факт, що звичайні космічні супутники передають дані, використовуючи тільки видимий спектр, тому ранкові прогнози виходять не дуже точними через відсутність накопичення інформації. Втім цей недолік можна компенсувати за рахунок зображень, отриманих в інфрачервоному випромінюванні. Набагато важливішим є той факт, що просторовий дозвіл геостаціонарних супутників достатній невеликий і становить близько 1 км – це набагато менше, ніж у знімків хмарності, що робляться із землі. Це призводить до того, що на фото відображаються великі конвективні хмари, у той час як дрібніші хмарні утворення можуть не відрізнятися – це призводить до зниження точності розрахунку інсоляції. Крім того, частота отримання/оновлення інформації (зображень) набагато менша, ніж за методом Total sky imagery. Варто також зазначити, що для обробки супутникових зображень знадобиться набагато більше часу, що погіршує точність прогнозів для невеликого (до 1 години) горизонту часу.

В цілому, за рахунок покриття великої площі хмарного покриття використання супутникових зображень дозволяє давати досить точний прогноз до 6 годин, який дуже часто перевершує дані числових методів прогнозу погоди.

#### **2.1.4 Прогнозування сонячної активності терміном до 1 доби**

Більшість прогнозів метеорологічної ситуації на тривалі, більше 1 доби, відрізки часу ґрунтуються на використанні чисельних прогнозів погоди (NWP), які активно використовуються практично у всіх сферах діяльності, де необхідно враховувати вплив природних факторів.

Сучасна модель NWP – це набір сучасних комп'ютерних програм, в яких за допомогою математичних та фізичних алгоритмів/рівнянь описані процеси, що відбуваються в атмосфері та характер їх змін із часом. Вихідні дані для розрахунків чисельних прогнозів погоди беруться з результатів та аналізу метеорологічних спостережень, при цьому чим точніше і якісніше вихідні дані (погодна обстановка), тим набагато точніше результат прогнозу. У цьому використовуються як дані для конкретної місцевості, на яку проводиться прогнозування розвитку погодної ситуації, а й дані спостереження верхніх верств атмосфери, зон океану і наземних зон.

Залежно від сфери застосування можуть використовуватися різні види моделей чисельних прогнозів погоди – від глобальних моделей (дають прогноз погоди на термін більше 15 діб) до мезомасштабних моделей, які дають можливість розраховувати короткочасні прогнози для відносно невеликих (з роздільною здатністю близько 15-100 км) регіонів та можливістю дуже частого оновлення/корекції отриманих результатів.

### **2.1.5 Просторово-часова інтерполяція та згладжування**

Оскільки чисельні прогнози погоди прораховуються окремих точок простору, їх використання у конкретній точці місцевості вимагає деякої форми інтерполяції. Найпростіший спосіб – це взяти результати прогнозів для двох найближчих точок та провести згладжування/усереднення результатів. Чим більше прогнозів для сусідніх точок, розташованих біля необхідної точки на місцевості, буде використано, тим точніший результат буде отримано в результаті. Існує кілька алгоритмів і математичних моделей, які дозволяють автоматично проводити згладжування отриманих результатів як щодо невеликих площ (близько 100 на 100 км), так досить великих районів (300 на 300 км). У районах, де активно

використовуються відновлювані джерела енергії (вітерові та сонячні електростанції), наприклад, для деяких регіонів Німеччини та Швейцарії, розроблені спеціальні додатки, що дозволяють проводити оптимальне згладжування для площ 10x10 км.

### **2.1.6 Погодинний прогноз**

Високі рівні проникнення розподілених енергетичних ресурсів (РЕМ), які зазвичай ґрунтуються на відновлюваній генерації, створюють низку проблем у роботі енергосистеми через властивий їм переривчастий та невизначений характер таких РЕМ. У цьому контексті дуже важливо розвивати здатність точно прогнозувати виробництво енергії з відновлюваних джерел, таких як фотоелектрична енергія для отримання короткострокових та середньострокових прогнозів. Точні прогнози забезпечують низку істотних переваг, а саме:

- **Можливість диспетчеризації:** щоденна робота безпечних енергосистем переважно залежить від диспетчеризації електростанцій на добу вперед. Відповідно, значущі плани на день уперед можуть бути виконані лише в тому випадку, якщо доступні точні прогнози вироблення електроенергії з відновлюваних джерел на день уперед разом із надійними прогнозами прогнозів споживання навантаження на день уперед;
- **Ефективність:** оскільки коливання вихідної потужності від переривчастих джерел можуть викликати коливання частоти та напруги в системі, деякі країни ввели штрафи для генераторів електроенергії, які не можуть точно спрогнозувати вироблення електроенергії наступного дня; Таким чином, деякі виробники енергії вважають за краще занижувати свої прогнози вироблення електроенергії на день вперед, щоб уникнути штрафів наступного дня. Така вимушена консервативна поведінка явно неефективна



- Моніторинг: невідповідності між прогнозами потужності та фактично виробленою потужністю можуть також використовуватись виробниками енергії для моніторингу роботи установки, оцінки природного зниження ефективності установки через старіння деяких компонентів або для раннього виявлення несправностей.

Сучасний стан і виробництво електроенергії на фотоелектричних установках в основному залежить від деяких метеорологічних змінних, таких як освітленість, температура, вологість або кількість хмар. З цієї причини прогнози погоди є загальним внеском у методології прогнозування виробництва фотоелектричних батарей. Залежно від конкретної розглянутої проблеми прогнози можуть також знадобитися в різних просторових і тимчасових масштабах, починаючи з високих тимчасових дозволів (тобто порядку хвилин) і дуже локалізованих (наприклад, прибережні вітряні електростанції) і закінчуючи грубішими тимчасовими дозволами (наприклад, годинник) ) та охоплюють розширену географічну область (наприклад, регіон або країну) для агрегованих завдань диспетчеризації електроенергії на день вперед. У той же час у літературі були розглянуті дуже різні підходи та методології, засновані на статистичних, математичних, фізичних, машинному навчанні або гібридних (тобто поєднують попередні) підходах.

Наприклад, використовує нечітку теорію для прогнозування інсоляції на основі даних про вологість та кількість хмар, а потім використовує рекурентну нейронну Мережі (RNNs) для прогнозування вироблення фотоелектричної енергії. Методи авторегресії (ARX) використовуються для короткострокових прогнозів (прогнози на хвилину вперед до двох годин вперед) з використанням просторово-часових моделей прогнозування сонячної радіації.

Також використовувався гібридний метод, заснований на погоді, де використовувалася картка (SOM), мережа квантування навчальних векторів (LVQ), метод регресії опорних векторів (SVR) і підхід нечіткого виведення був об'єднаний разом для прогнозування вироблення електроенергії для однієї фотоелектричної установки. В екстремальному навчанні машини (ELMS) використовуються для прогнозування вироблення електроенергії експериментальною фотоелектричною системою в Шанхаї. Глобальний конкурс із прогнозування енергетики (GEFCom2014) дозволив порівняти різні алгоритми у конкурентний спосіб вирішення завдань ймовірного прогнозування енергії для детального опису результатів конкурсу. GEFCom2014 складався з чотирьох треків, присвячених прогнозуванню навантаження, цін, вітру та сонячної енергії.

В останньому випадку мета полягала в тому, щоб спрогнозувати вироблення сонячної енергії на постійній основі на 24 години вперед для трьох сонячних електростанцій, розташованих у певному регіоні Австралії (точне місце сонячних електростанцій не було розкрито учасникам конкурсу).

На продуктивність різних моделей прогнозування впливають багато елементів невизначеності, і не завжди зрозуміло, як окремі варіанти (наприклад, вибір конкретної методології прогнозування в порівнянні з іншою) або різні фактори сприяють остаточній помилці прогнозування (тобто з точки зору прогнозованого порівняно з фактичною вироблення електроенергії). Фактично більшість існуючих відповідних документів пропонують єдину методологію для виконання завдання прогнозування потужності та порівняння їх результатів з іншими дуже простими алгоритмами, у той час як порівняння між різними складнішими підходами може бути нелегко виконане. Фактично різні автори працювали з різними

наборами даних, і остаточні результати не можна порівнювати, оскільки вони залежать від конкретного періоду року, протягом якого було обчислено помилку прогнозування (наприклад, у зимовий час помилка зазвичай нижча, враховуючи, що кількість ненульових годин виробництва електроенергії нижча); похибка також залежить від конкретної країни (наприклад, простіше прогнозують сонячний день літнього часу в Італії, ніж в інших країнах з більш нестабільною погодою); і в цілому використання різних метрик помилок, різних інструментів прогнозування погоди з різними точністю, різних розмірів фотоелектричних установок і різних технологій або об'єктів все також сприяють визнанню недійсним простого порівняння різних методик прогнозування на підставі лише остаточних результатів точності.

### **2.1.7 Види тимчасових рядів**

Тимчасовим рядом називається впорядкована за часом послідовність спостережень. Виділяють одновимірні часові ряди, отримані при фіксованій кількісній характеристиці, і багатовимірні часові ряди, отримані при спостереженні кількох характеристик виділеного об'єкта. Тимчасові ряди можуть бути дискретними та безперервними. У складі дискретних виділяють ряди для рівновіддалених моментів спостереження і довільних моментів спостереження. Тимчасові ряди бувають детермінованими та випадковими: перші одержують на основі значень деякої не випадкової функції; другі є результатом реалізації деякої випадкової величини. Особливо слід виділити стаціонарні та нестаціонарні ряди. Ряд  $y(t)$  називається стаціонарним (у вузькому значенні), якщо закон розподілу ймовірностей випадкової величини  $y(t)$  не залежить від  $t$ .

### **2.1.8 Регресійні моделі для виділення тренду**

Оскільки під трендом розуміється деяка стійка і довготривала тенденція зміни, компоненту, відповідну тренду, можна наблизити поліномом від деякою мірою  $t$ . Нехай, як і раніше, задано ряду  $y_1, y_2, \dots, y_t, \dots, y_n$ . Тоді можна для отримання уявлення про тренд покласти:

$$y_t = a_0 + a_1 t + \dots + a_p t^p.$$

Поліном першого ступеня відбиває рівномірне у часі зростання чи спадання значень ряду. Поліном другого ступеня може виражати тенденцію зростання і подальшого зменшення значень ряду або навпаки і т.д. Зазвичай  $p$  мало проти  $n$ . Для отримання оцінок коефіцієнтів у формулі (1) можна використовувати техніку регресійного аналізу. Питання ступеня полінома вирішується кожної завдання окремо з урахуванням перевірки гіпотез про рівність нулю коефіцієнтів при старших ступенях у формулі (1) до того часу, поки гіпотеза про рівність нулю коефіцієнта  $a_p$  відхиляється. Іноді у тимчасових рядах проявляються тренди, які описуються функціями, нелінійними за параметрами, що підлягають оцінці: експоненційною

$$y_t = ak^t, \text{ де } k - \text{ темп зміни в разях, логарифмічної } y_t = a_0 + a_1 \log t$$

ступінною –  $y_t = a_0 t^{a_1}$ , гіперболічною  $y_t = a_0 + \frac{a_1}{t}$ , логістичною  $y_t = k/(1 + ae^{-\lambda t})$  та ін. Регресійний аналіз у цій ситуації проводиться за допомогою методу найменших квадратів після лінеаризуючих перетворень або нелінійного методу найменших квадратів.

### 2.1.9 Модель ARIMA

Модель ARIMA (Autoregressive Integrated Moving Average) – один із найпоширеніших методів аналізу та прогнозування часових рядів. Ця модель дозволяє обробити дані тимчасового ряду, щоб краще зрозуміти цей ряд або передбачити його розвиток.

ARIMA використовує три основні параметри ( $p$ ,  $d$ ,  $q$ ), які виражаються цілими числами. Тому модель також записується як ARIMA ( $p$ ,  $d$ ,  $q$ ). Разом ці три параметри враховують сезонність, тенденцію та шум у наборах даних:

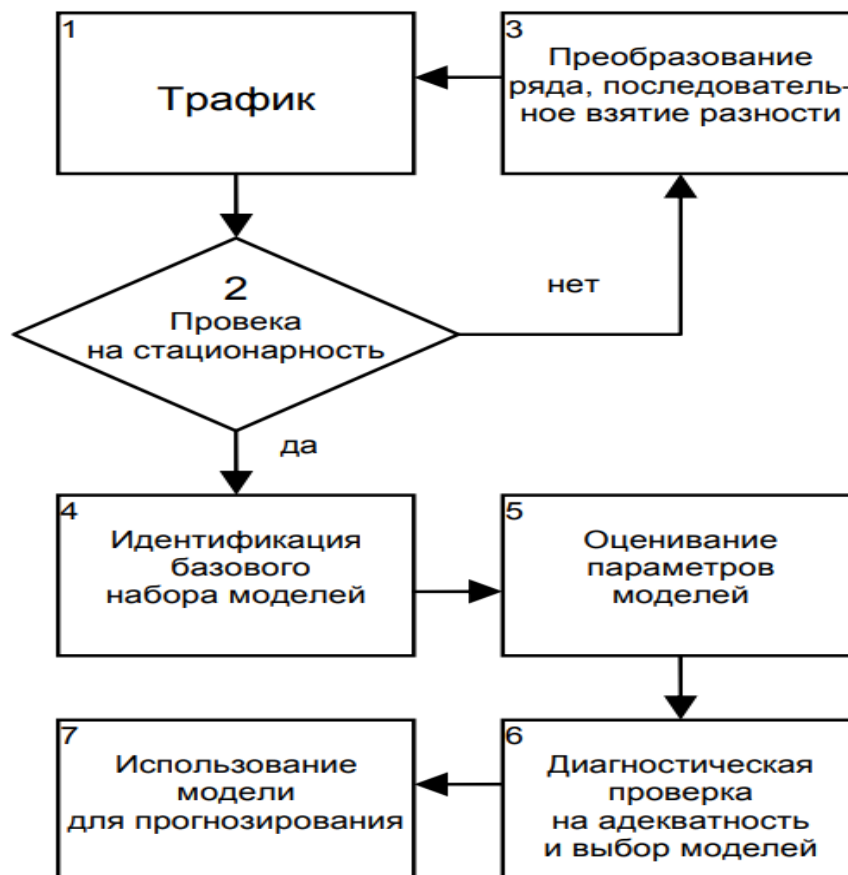
$p$  – порядок авторегресії (AR), який дозволяє додати попередні значення часового ряду. Цей параметр можна проілюструвати твердженням «завтра, ймовірно, буде тепло, якщо останні три дні було тепло».

- $d$  – порядок інтегрування (I; т. е. порядок різниць вихідного часового ряду). Він додає в модель поняття різниці часових рядів (визначає кількість минулих часових точок, які потрібно відняти від поточного значення). Цей параметр ілюструє таке твердження: "завтра, ймовірно, буде така сама температура, якщо різниця в температурі за останні три дні була дуже мала".

- $q$  – порядок ковзного середнього (MA), який дозволяє встановити похибку моделі як лінійну комбінацію значень помилок, що спостерігалися раніше.

Для відстеження сезонності використовується сезонна модель ARIMA – ARIMA( $p,d,q$ )( $P,D,Q$ ) $s$ . Тут ( $p$ ,  $d$ ,  $q$ ) – несезонні параметри, описані вище, а ( $P$ ,  $D$ ,  $Q$ ) слідує тим самим визначенням, але застосовуються до сезонної складової часового ряду. Параметр  $s$  визначає періодичність часового ряду (4 – кварталні періоди, 12 – річні періоди тощо).

Методологія побудови ARIMA-моделі для тимчасового ряду, що досліджується, включає наступні основні етапи [1]: - ідентифікацію пробної моделі; - оцінювання параметрів моделі та діагностичну перевірку адекватності моделі; - Використання моделі для прогнозування. Таким чином, спочатку (у блоці 1-3) необхідно одержати стаціонарний ряд. На цьому етапі рекомендується проводити аналіз автокореляційної функції (АКФ) та приватної автокореляційної функції (ЧАКФ). Швидке згасання значень АКФ – простий тест на стаціонарність. На цьому етапі використовуються також статистичні тести на наявність одиничного кореня (розширений тест ДікіФуллер або ADF-тест) [1]. Якщо відповідно до статистики ДікіФуллера чи оцінок АКФ ряд є нестаціонарним, то переходу до стаціонарного ARIMA – модель прогнозування значень трафіку традиційно застосовують оператор взяття послідовних різниць, цим визначається значення параметра  $d$  (порядку різниці).



Таким чином, значення одного параметра моделі ARIMA(p,d,q) вже відомо. У блоці 4 після отримання стаціонарного ряду досліджується характер поведінки вибіркового АКФ і ЧАКФ і висувуються гіпотези про значення параметрів  $p$  (порядок авторегресії) і  $q$  (порядок ковзного середнього). На вході блоку 4 може формуватися базовий набір, що включає одну, дві або навіть більше моделей, іншими словами, портфель моделей. У блоці 5 після здійснення ідентифікації моделі необхідно оцінити параметри. Для цього використовується метод максимальної правдоподібності (ММП). У блоці 6 для перевірки кожної пробної моделі адекватність аналізується її ряд залишків. У адекватної моделі ряд залишків може бути схожий білий шум, тобто. їх вибіркові АКФ не повинні відрізнятися від нуля. Для перевірки гіпотези про те, що дані є реалізацією «білого шуму», використовується також Q-статистика.

### 2.1.10 Тета-модель

Модель ґрунтується на концепції модифікації локальних викривлень часового ряду. Ця зміна виходить за допомогою коефіцієнта, званого тета-коефіцієнта (як символ використовується грецька літера Тета), який застосовується безпосередньо до інших різниць часового ряду:

$$X''_{new}(\theta) = \theta * X''_{data} \quad \text{где } X''_{data} = X_t - 2X_{t-1} + X_{t-2} \text{ при часі } t$$

Якщо локальні викривлення поступово зменшуються, то тимчасовий ряд  $1c$  зменшується. Чим менше значення коефіцієнта тета, тим більший ступінь дефляції. У граничному випадку, коли  $\theta=0$ , часовий ряд  $1c$  перетворюється на лінію лінійної регресії і поступове зменшення коливань зменшує абсолютні відмінності між послідовними термінами у похідному ряду і є в якісному відношенні пов'язаний з появою довгострокових тенденцій даних (Assimakopoulos, 1995).

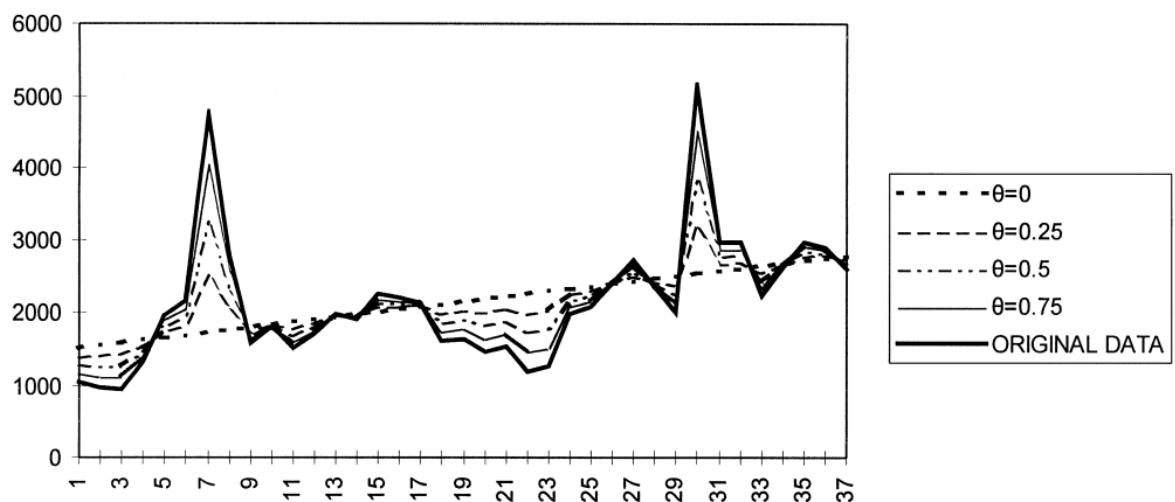


Рис.2.1

Коефіцієнт може приймати і негативні значення, але в даному контексті вони не представляють інтересу і далі не розглядаються і навпаки, якщо локальна кривизна збільшується ( $\theta > 1$ ), то тимчасовий ряд розширюється. Чим більший ступінь розширення, тим більше збільшення короткострокової поведінки.

Дотримуючись цієї процедури, можна отримати набір нових тимчасових рядів, про тета-линій.

Розміщення цих ліній по відношенню до вихідних даних може бути виконано у різний спосіб. Якщо припасування є процедурою оцінки OLS, то середнє значення і нахил тета-ліній залишаються незмінними в порівнянні з тими вихідними даними. Загальне формулювання методу виглядає так: вихідний



часовий ряд розкладається на дві або більше тета-ліній. Кожна з тета-ліній екстраполюється окремо, а прогнози просто об'єднуються. Будь-який метод прогнозування може бути використаний для екстраполяції тета-лінії відповідно до існуючого досвіду (Fildes, Hibon,

Makridakis & Meade, 1998). Для кожного об'єкту прогнозування можна використовувати різні комбінації тета-ліній, для кожного об'єкту прогнозування. Це можна продемонструвати з прикладу однієї з найпростіших випадків, у яких вихідний часовий ряд розкладається на дві тета-лінії, тобто.  $\theta=0$  і  $\theta=2$ :

$$\text{Data} = 1/2 (L\theta = 0) + L(\theta = 2)$$

де  $L(\theta = 0)$  позначає тета-лінію для  $\theta$

параметра, що дорівнює нулю.

Перша тета-лінія ( $\theta = 0$ ) є лінією регресії даних, а друга має другі різниці рівно вдвічі більше вихідного часового ряду. Це випадок, коли дві, екстремальні та симетричні щодо 1, тета-лінії складені. Перша компонента  $L(\theta = 0)$  описує часовий ряд через лінійний тренд. Друга,  $L(\theta=2)$ , подвоює локальні викривлення, збільшуючи короткострокову поведінку. Перша тета-лінія екстраполюється звичайним способом для лінійного тренду. Друга екстраполюється за допомогою простого експоненційного згладжування.

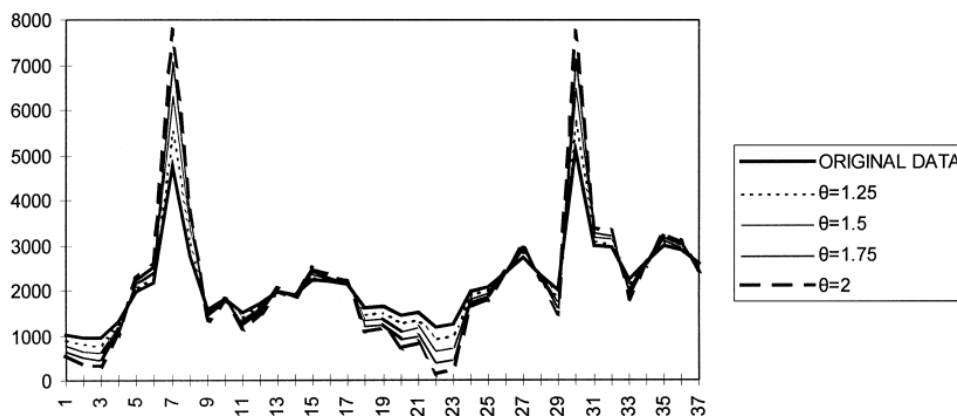


Рис.2.2

### 2.1.11 Використання штучних нейронних мереж

НС має на увазі мінімальну участь аналітика у формуванні моделі тимчасового ряду, так як здатність нейромережевих моделей до навчання дозволяє виявити приховані взаємозв'язки та закономірності між даними, а алгоритми навчання адаптують вагові коефіцієнти відповідно до структури даних, представлених для навчання. Використання апарату СР для прогнозування ВР з полягає у формуванні СР певної структури, в її параметричному налаштуванні на основі поведінки досліджуваної системи в заздалегідь відомі моменти часу, в передбачанні майбутньої поведінки системи за її передісторією. Вибір структури нейронної мережі обумовлюється специфікою і складністю задачі, що розв'язується.

Для вирішення деяких типів завдань розроблено оптимальні конфігурації. Параметричне налаштування СР відбувається на основі навчання мережі на виділеній з безлічі вихідних даних навчальній вибірці (ОВ). Під час навчання з учителем кожному за навчального вхідного прикладу потрібно знання правильної відповіді. Нейронної мережі пред'являються значення вхідних та вихідних сигналів, а вона за певним алгоритмом підлаштовує ваги синаптичних зв'язків. У процесі навчання проводиться коригування ваги мережі за результатами порівняння фактичних вихідних значень з відомими заздалегідь і очікуваними вихідними значеннями. Під час навчання СР відбувається зміна вагових коефіцієнтів мережі виходячи з зміни фактичної похибки на ітераціях навчання. Якість навченої СР оцінюється на контрольній вибірці (КВ), також виділеної з безлічі вихідних даних. ОВ і КВ не перетинаються і, як правило, у багатьох вихідних даних хронологічно слідує один за одним. Якщо на КВ значення помилки знаходиться в допустимих межах, то налаштування СР вважається завершеним, а

нейронна мережа готової на вирішення завдання прогнозування. При побудові моделі прогнозування часто доводиться виконувати деяке попереднє перетворення даних із задоволення ряду відомих вимог. Це стосується як прогнозованої величини (ПВ), так і навчальної та контрольної вибірок. Основна вимога до ПВ полягає в тому, щоб можна прогнозувати майбутні значення тимчасового ряду. Як правило, залежно від завдання, доводиться вдаватися до таких методів перетворення вхідних даних, які дозволяють правильно судити про закономірності і особливості в даних, що відображають їх якісні характеристики. Часто постає завдання відображення даних у простір меншої розмірності. Така згортка даних призводить до усунення надмірності даних і скорочення часу навчання СР.

Однією з головних вимог у часовому ряду є його стаціонарність, яка полягає в тому, що розподіл його значень інваріантний щодо моменту часу, для якого він побудований. Для характеристики стаціонарності використовується те, що для двох вибірок, побудованих у різні моменти часу, закон розподілу має залишатися тим самим. Якщо елемент аналізованих часових рядів багатовимірний, слід аналізувати кожен компонент  $x_i$  елемента часового ряду і перевірити, що він має рівномірний розподіл на відрізку  $[m_i - \sigma_i, m_i + \sigma_i]$ , де  $m_i$  - математичне очікування ознаки  $x_i$ ,  $\sigma_i$  - її дисперсія.

Важливою передумовою для правильного відновлення інтерполюваної функції нейронною мережею є те, щоб набори, що входять до ВВ та КВ не суперечили один одному. Це означає, що двох близьких векторів вибірки значення прогнозованої величини також близькі. На практиці застосовується підхід, який полягає в тому, що два вектори вважаються близькими, якщо один знаходиться в околиці іншого. Кордони околиці вибирають так, щоб умова мала місце для 50-100% наборів из ОВ или КВ.

Таким чином, для забезпечення якості побудованої нейронної мережі слід відповідним чином вибрати дані для навчання, так і дані для її оцінки.

Від навчальної вибірки залежить також час, необхідний навчання мережі.

Навчальна вибірка має бути представницькою. Існують різні евристики на вибір “ковзного вікна”, що оглядає ділянку тимчасового ряду, що використовується як вхід нейронної мережі. Оскільки нейронна мережа працює тільки з числовими вхідними даними, важливим етапом при підготовці даних є перетворення та кодування даних. Дані, призначені для навчання, повинні бути нормалізовані та їх значення мають бути розподілені у певному діапазоні

## **2.2 Моделі прозорого, чорного та сірого ящика.**

Існує три підходи до тестування програмного забезпечення: тестування білої, сірої та чорної скриньок. Кожен їх розглядає процес з іншого погляду, і може бути використаний як єдиний підхід. Завершення всіх трьох етапів гарантує якість продукту.

### **2.2.1 Тестування білої скриньки**

Головна мета цього процесу – перевірка коду, тестування як внутрішньої структури, так і дизайну. Тестувальники можуть бачити код на цій стадії, тому цей метод має ряд інших назв, таких як відкрите тестування або перевірка коду. Кожен термін вказує на прозорий процес, як і можливість перевірки процесів введення-виведення, ґрунтуючись на внутрішніх алгоритмах роботи.

Тестування білої скриньки передбачає пошук та покращення наступних моментів:

- Неробочих або неоптимізованих ділянок коду
- Втрата безпеки
- Робочі процеси та сценарії введення

- Умовні процеси
- Неправильне функціонування об'єктів
- Некоректне відображення інформації.

### 2.2.2 Тестування чорного ящика

Під час поведінкового тестування або тестування чорної скриньки фахівець не знає напевно, що за продукт він тестує. Внутрішня структура, додаток та дизайн залишаються невідомими для тестувальника. Тести, зазвичай, функціональні. Веб-сторінки вивчаються з використанням браузера, введенням деяких даних та під час аналізу отриманих даних.

Метод чорної скриньки застосовується на наступних рівнях тестування системи

- Тестування інтеграції
- Приймальні випробування

Кількість методів тестування залежить від складності препарату, тобто. ящика.

Підходи до розробки алгоритмів тестування чорної скриньки бувають такі :

- Причинно-наслідкові (визначення випадків та їх вплив на систему)
- Аналіз крайніх значень (визначення меж введення)
- Розмітка еквівалентності (дійсні та недійсні розмітки)

### 2.2.3 Методи тестування програми як "білої скриньки" та "чорної скриньки"

При тестуванні білої скриньки (англ. white-box testing, також кажуть - прозорі скриньки), розробник тесту має доступ до вихідного коду і може писати код, пов'язаний з бібліотеками ПЗ, що тестується. Це типово для юніт-

тестування (англ. unit testing), у якому тестуються лише окремі частини системи. Воно забезпечує те, що компоненти конструкції - працездатні та стійкі, до певної міри.

При тестуванні чорної скриньки (англ. black-box testing), тестувальник має доступ до ПЗ тільки через ті ж інтерфейси, що і замовник або користувач, або через зовнішні інтерфейси, що дозволяють іншому комп'ютеру або іншому процесу підключитися до системи для тестування. Наприклад, модуль, що тестує, може віртуально натискати клавіші або кнопки миші в тестованій програмі за допомогою механізму взаємодії процесів, з впевненістю в тому, чи все йде правильно, що ці події викликають той же відгук, що і реальні натискання клавіш і кнопок миші. Як правило, тестування чорної скриньки ведеться з використанням специфікацій або інших документів, що описують вимоги до системи.

#### **2.2.4 Тестування сірої скриньки**

Тестування сірої скриньки передбачає часткову поінформованість про внутрішні процеси. Даний метод – це комбінація двох попередніх підходів (тестування білої та чорної скриньок). Фахівець прагне знайти всі проблеми функціонування та помилки в коді. На цій стадії тестувальник може продати наскрізний тест.

Підходи до тестування:

- Тестування шаблонів
- Тестування матриці
- Регресійне тестування
- Тестування з використанням ортогонального масиву

Для запуску тестових випадків під час тестування сірої скриньки не є обов'язковим доступ до коду. Кожен тест базується на знанні поведінки програми. Це добрий підхід до реалізації функціонального тестування. Однак, це не буде успішним без реалізації більш глибоких методів, таких як тестування білого та чорного ящиків.

Мережі з довготривалою короткочасною пам'яттю (LSTM) - це особливі рекурентні нейронні мережі (PHC).

Мережі (RNNs), які були вперше представлені в [17]. PHC - це нейронна мережа (НС) з рекурентними зв'язками між нейронами, що дозволяє їй вчитися на основі поточної та попередньої інформації для пошуку кращого рішення. Однак, коли два осередки в RNN знаходяться далеко один від одного, важко отримати корисну інформацію через проблеми зникнення градієнта та вибуху. Вирішенням цієї проблеми є спеціальні нейрони, які називаються осередками пам'яті. Використовуючи спеціальні нейрони, LSTM може зберігати корисну інформацію протягом довільного періоду часу. Більше того, осередки LSTM мають здатність дізнаватися, які дані необхідно зчитувати, зберігати та прати з пам'яті шляхом регулювання трьох різних керуючих затворів, а саме затвора забування  $f_t$ , вхідного затвора  $i_t$  та вихідного затвора  $o_t$ , як показано на малюнку 2. Затвор забування  $f_t$  вирішує, зберігати або відкидати інформацію зі стану комірки, як показано в рівнянні (2). Логістична функція генерує або 0, або 1 як значення для  $f_t$ , что означает либо отказ, либо сохранение текущего состояния клетки на временном шаге  $t$ , соответственно. Где  $w$ ,  $h$ ,  $x$  и  $b$  - вес, выход, вход и смещение, соответственно. Функция активации представлена  $\sigma$ . С другой стороны, входные ворота управляют тем, які вхідні значення можуть бути збережені в стані клітини, як показано в рівнянні 4. Де  $i_t$  являє собою сигнал (0 або 1), який управляє процедурою оновлення,  $g_t$  - нове значення кандидата, а  $s_t$  - новий стан комірки. Більше того, вихідні ворота  $o_t$  відповідають за передачу збереженої інформації наступним

нейронам, як показано в рівнянні 6. Сонячна енергія генерація прогнозування. Таким чином, LSTM є потужною ML-моделлю, що дозволяє вловлювати довгострокову залежність, а також нелінійну залежність.

### 2.3 Алгоритми ансамблевого прогнозування

Ансамбль моделей прогнозування – це ключове покращення точності прогнозування, яке привернула увагу багатьох дослідників останніми роками. У нашому дослідженні ми запропонували ML та статистичну гібридну модель (MLSHM), яка поєднує сонячну фотоелектричну потужність, передбачену на основі різних моделей.

ML моделей та статистичної моделі. Більш того, кілька методів ансамблю були використані для поєднання прогнозів різних моделей та генерування остаточного прогнозу сонячної фотоелектричної потужності. Щоб збільшити та підвищити переваги поєднання ML моделей зі статистичним методом, забезпечення різноманітності між об'єднаними моделями є важливою процедурою. У цьому дослідженні різноманітність в ансамблевих методах відноситься до двох категорій:

- (i) різноманітність даних та
- (ii) структурна різноманітність.

Різнманітність даних досягається шляхом створення кількох наборів даних із вихідного набору даних для навчання ML-моделей [25,32]. Структурне розмаїття досягається з допомогою різних архітектур моделей прогнозування [32]. У нашому дослідженні ми запровадили різноманітність даних у ML-моделі та структурне розмаїття шляхом поєднання двох різних за структурою алгоритмів, тобто. ML-моделі та статистичної моделі. В результаті різноманітність даних було застосовано до об'єднаних ML-моделей наступним чином:



1. Після того, як набір даних був розділений на навчальний та тестовий набори, навчальний набір був

далі ділиться на  $n$  навчальних підмножин, де нашому дослідженні  $n = 5$ .

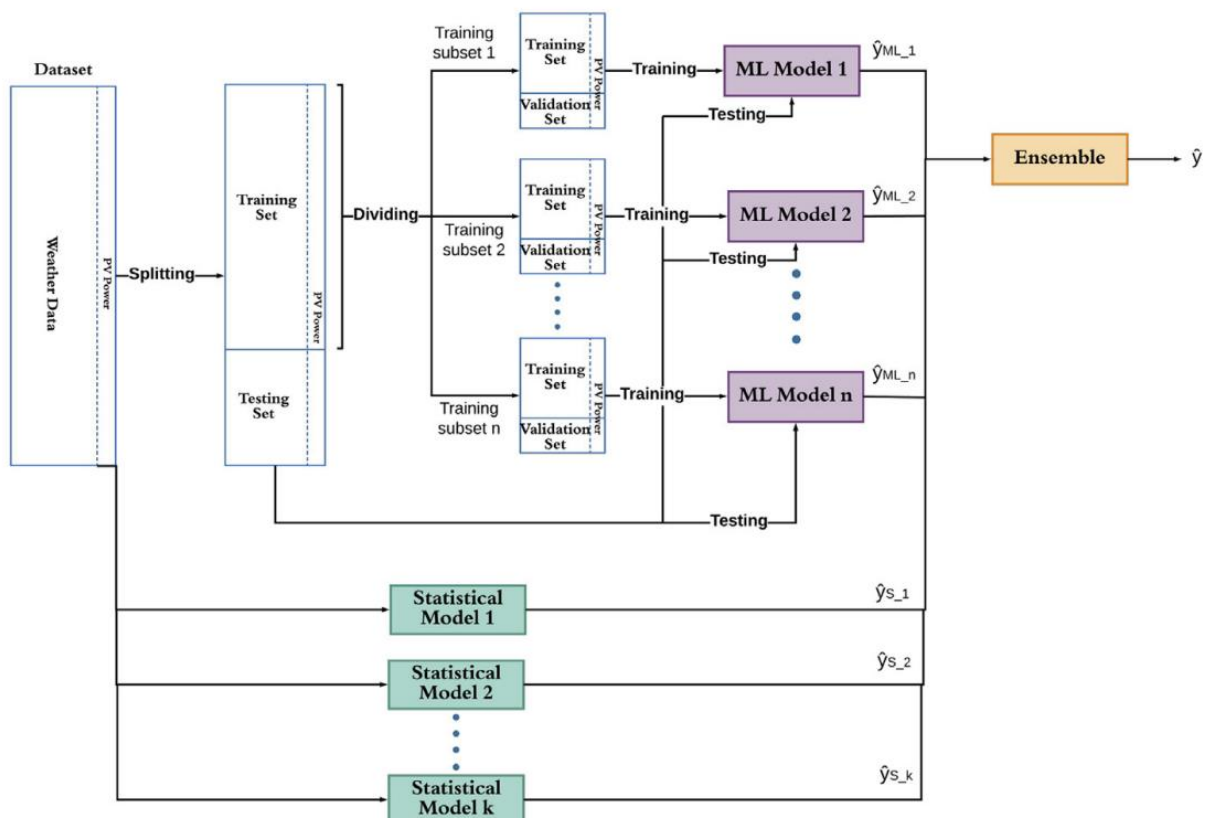
2. Кожна ML-модель навчається на одному з навчальних підмножин; таким чином, ми маємо  $n$  ML-моделей

навчених на різних наборах даних, де в нашому дослідженні  $n = 5$ , отже, ми маємо дві моделі ML

моделей, навчених різних наборах даних.

3. Всі моделі ML тестуються на тому самому наборі даних для порівняння, щоб

забезпечити рівність результатів між моделями.



блок-схема запропонованої MLSHM, що показує різноманітність даних, що застосовуються до навчального набору. Є  $n$  моделей ML та  $k$  статистичних методів, що прогнозують сонячну ФЕМ.

На першому етапі ML-моделі та статистичні моделі прогнозували сонячну фотоелектричну потужність окремо, а потім ми об'єднали ці результати для отримання остаточного прогнозу. Ми досліджували чотири різні методи ансамблю, щоб перевірити їх ефективність на запропонованій MLSHM. Методи ансамблю описані таким чином:

1. EN1: метод простого усереднення, який є найпростішим і найприроднішим методом, що генерує остаточний прогноз сонячної фотоелектричної енергії шляхом взяття середнього значення прогнозів, отриманих за допомогою ML-моделей та статистичних моделей. Підсумкова сонячна фотоелектрична потужність генерується так:

$$y = \sum_{j=1}^m y_j / m$$

Тут  $m$  означає кількість членів ансамблю, яке дорівнює  $m=n+k$ , а  $y$  являє собою прогнозовану сонячну ФЕМ,  $y_j$  представляє прогнозовану сонячну фотоелектричну потужність члена ансамблю  $j$ , а  $y$  - остаточний прогноз сонячної фотоелектричної потужності.

2. EN2: зважене усереднення з використанням лінійного підходу, де кінцева сонячна потужність досягається шляхом присвоєння різних ваг об'єднаним моделям на основі їх точності. Відповідно, моделі з більш високим коефіцієнтом помилок присвоюється менша вага, так що вона вносить мінімальний внесок у остаточний прогноз. І навпаки, моделі з меншим коефіцієнтом помилок (точнішою) присвоюється більша вага; таким чином, вона робить більший внесок у остаточний прогноз. Ваги розраховуються таким чином, щоб сума ваги всіх моделей ансамблю дорівнювала 1.

$$\omega_i = 1 - \frac{nMAE_i}{\sum_{j=1}^m (1 - nMAE_j)}$$

$$y = \sum_{i=1}^m y_i * \omega_i$$

Тут  $w$  є вагою, присвоєною моделі  $i$ , а  $nMAE_i$  - нормалізоване середнє значення абсолютної помилки моделі  $i$

3. EN3: виважене усереднення з використанням нелінійного підходу, де використовується та ж концепція, що й EN1, але ваги моделей розраховуються як функція softmax від негативної величини помилки ( $nMAE$ ):

$$\omega_i = \exp^{-nMAE_i} / \sum_{j=1}^m \exp^{-nMAE_j}$$

Тут  $\exp$  означає експоненційну функцію. Остаточний прогноз сонячної енергії розраховується за попередніми обчисленнями

4. EN4: комбінація через дисперсію з використанням інверсного підходу – це просто зважене усереднення з використанням наступного рівняння зважування:

$$\omega_i = 1 / nMAE_i / \sum_{j=1}^m 1 / nMAE_j$$

Остаточний прогноз сонячної фотоелектричної потужності розраховується шляхом виваженого усереднення, як показано на другому прикладі.

## 2.4 Основна ідея дослідження

Наведений вище короткий огляд проблем моделювання фотоелектричних процесів та підходів до їх вирішення свідчить про актуальність досліджень прогностичних моделей, робастних щодо помилок метеорологічної бази даних, що використовується для їхнього навчання. У цій роботі ставиться мета розробки своєї методології побудови таких прогностичних моделей, заснованої на ідеї рефлексивного навчання

## Розділ 3

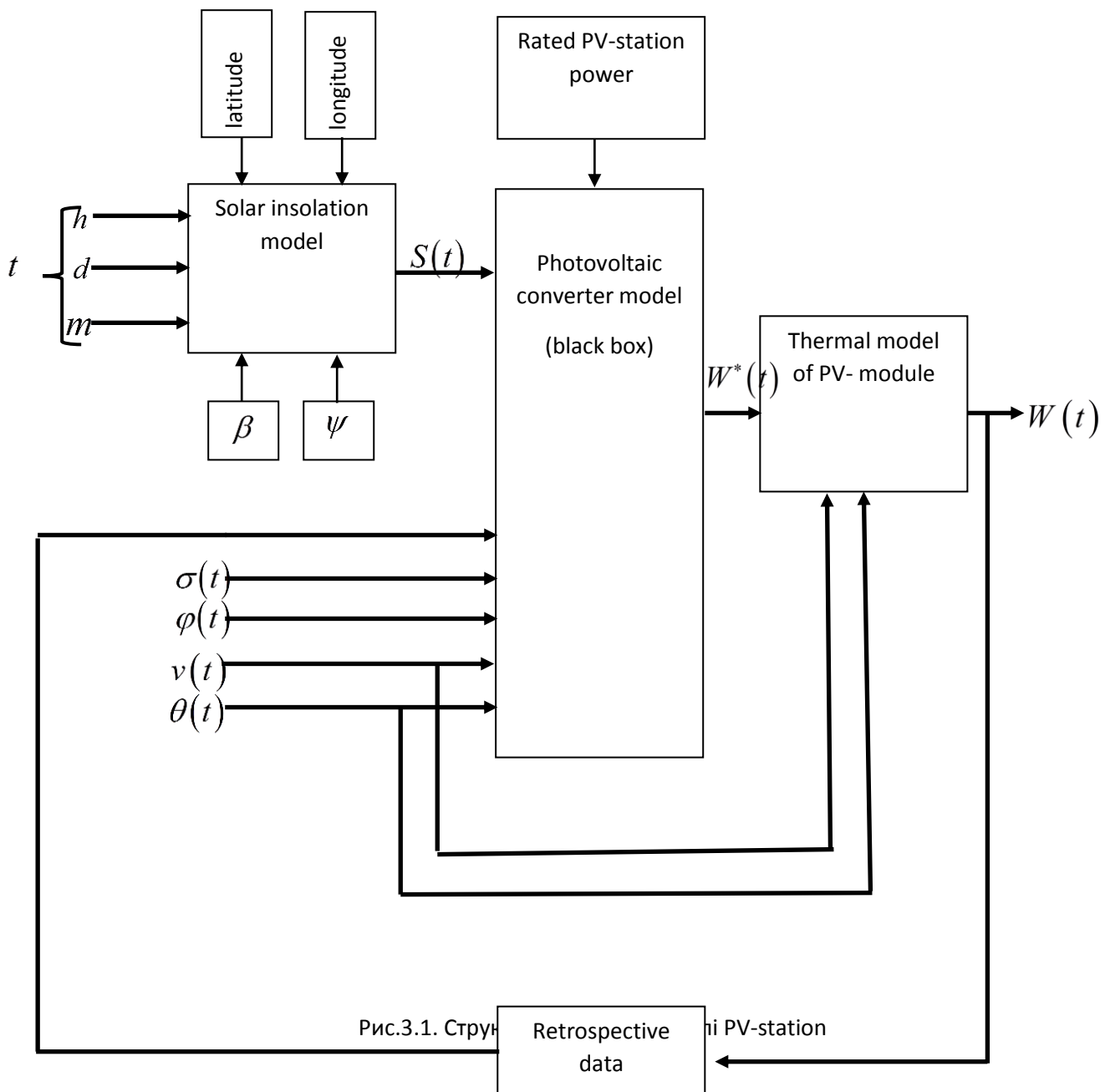
## РОЗРОБКА ТА ДОСЛІДЖЕННЯ МАТЕМАТИЧНОЇ МОДЕЛІ СИСТЕМИ ПРОГНОСТИЧНОГО МОНІТОРИНГУ ФОТОЕЛЕКТРИЧНОЇ СТАНЦІЇ

### 3.1 Концепція побудування математичної моделі процесу генерації сонячної енергії.

Концептуальну схему, розвиваємого в даній роботі підходу до моделювання фотоелектричної станції, можливо охарактеризувати наступними класифікаційними ознаками: екзогенна модель по типу «сірого ящика» (рис 3.1). Ідеальний рівень  $S(t)$  сонячної інсоляції (включаючи пряму та дифузну компоненти) в умовах «чистого неба» і пропорційна їй кількість електроенергії, що виробляється в цих ідеальних умовах, визначаються детермінованим алгоритмом функції від моменту часу (година, день, місяць), що характеризує взаємне розташування Сонця, Землі, умовної точки на її поверхні, що вказує місце розташування фотоелектричної станції, що моделюється, а також у функції кутів, під якими падають сонячні промені на фотоелектричні панелі (рис 3.2). Реальна кількість електроенергії  $W(t)$ , виробляється фотоелектричною станцією, завжди менше цього ідеального рівня і залежить від безлічі метеорологічних факторів, що впливають. У моделі враховуються: температура повітря  $\theta(t)$  і температура PV-модуля  $\Theta(t)$ , вологість повітря  $\varphi(t)$ , швидкість вітру  $v(t)$ , хмарність  $\sigma(t)$ . Характерною особливістю фотоелектричної станції являється те, що всі ці метеорологічні фактори впливають на рівень виробленої електроенергії мультиплікативно. При їх зміні, кількість виробленої електроенергії може зменшитися або збільшитися в відсотковому відношенні, но не може перевищувати ідеальний рівень або стати від'ємним. Зміна потужності фотоелектричної станції під впливом метеорологічних факторів відбувається внаслідок зміни коефіцієнта перетворення енергії, а не в

результаті сумми енергії сонця з якоюсь додатковою енергією з інших джерел.

В цьому сенсі метеорологічними факторами обумовлена деяка змінна резистивність моделі по відношенню до вхідної сонячної енергії.



## 2.2. Модель фотоелектричного «чорного ящика»

Ігноруючи реальну фізичну природу фотоелектричного перетворення, представимо характеризуюче його відношення «вхід - вихід» в вигляді

$$W^*(t) := S(t) \cdot F(\sigma, \varphi, \nu, \theta, h, d), \quad (1)$$

де  $W^*(t)$  - генеруюча станцією електроенергія в докритичному температурному діапазоні, вище котрого відбувається аномальне зниження рівня генерації,  $F(\sigma, \varphi, \nu, \theta, h, d)$  - функція фотоелектричного перетворення, враховуюча залежність його ефективності від погодних умов в місці розположення PV-станції і циклічного часу, представлено номером години  $h$  впродовж дня і номером дня  $d$  впродовж місяця .

Задача побудови моделі фотоелектричного «чорного ящика» є в тому, щоб найбільш точно виразити в явному вигляді залежність функції

$F(\sigma, \varphi, \nu, \theta, h, d)$  від її аргументів на основі ретроспективних даних щодо величини сонячної інсоляції  $S(t)$ , генерованої електроенергії  $W^*(t)$  та врахованих метеорологічних даних, отриманих в період навчання моделі.

Наш підхід до рішення цієї задачі опирається на гіпотезу, згідно котрої функція  $F(\sigma, \varphi, \nu, \theta, h, d)$  може бути факторизована и представлена в вигляді множини обособлених функцій впливу

$$F(\sigma, \varphi, \nu, \theta, h, d) := f_1(\sigma) \cdot f_2(\varphi) \cdot f_3(\nu) \cdot f_4(\theta) \cdot f_5(h) \cdot f_6(d). \quad (2)$$

Далі кожену функцію впливу представимо в вигляді експоненти від відповідного полінома ступіня  $k$

$$\begin{aligned}
f_1(\sigma) &= \exp\{a_{10} + a_{11}\sigma^1 + a_{12}\sigma^2 + \dots + a_{1k}\sigma^k\}, \\
f_2(\varphi) &= \exp\{a_{20} + a_{21}\varphi^1 + b_{22}\varphi^2 + \dots + b_{2k}\varphi^k\}, \\
f_3(v) &= \exp\{a_{30} + a_{31}v^1 + a_{32}v^2 + \dots + a_{3k}v^k\}, \quad (3) \\
&\dots\dots\dots \\
f_6(d) &= \exp\{a_{60} + a_{61}d^1 + a_{62}d^2 + \dots + a_{6k}d^k\}.
\end{aligned}$$

При навчанні моделі підбираються такі значення коефіцієнтів  $a_{ij}$ , при котрих мінімізується середньоквадратичний небаланс наближеної рівності (1).

Логарифмуючи праву та ліву частини рівності (1), представимо мінімізуємий функціонал в поліноміальному вигляді

$$\sum_t \left\{ \ln \left( \frac{W^*(t)}{S(t)} \right) - \ln [F(\sigma, \varphi, v, \theta, h, d)] \right\}^2 = \min. \quad (4)$$

Враховуючи умови досягнення екстремума

$$\frac{\partial}{\partial a_{ij}} \left[ \sum_t \left\{ \ln \left( \frac{W^*(t)}{S(t)} \right) - \ln [F(\sigma, \varphi, v, \theta, h, d)] \right\}^2 \right] = 0, \quad (5)$$

Отримуємо систему  $6 \cdot k$  лінійних рівнянь відносно шуканих коефіцієнтів  $a_{ij}$ , якими визначаються значення функції фотоелектричного перетворення  $F(\sigma, \varphi, v, \theta, h, d)$ .

Прогнозоване значення  $\hat{W}^*(t)$  вироблення електроенергії з урахуванням знайдених значень коефіцієнтів  $a_{ij}$  дорівнює

$$\hat{W}^*(t) = S(t) \cdot F(\sigma, \varphi, v, \theta, h, d). \quad (6)$$

### 2.3. Теплова модель PV – модуля

Рівняння теплового балансу PV – модуля в умовах конвективного теплообміну з довкіллям можна подати у вигляді

$$cm \frac{dT_M^0}{dt} = S(t) \frac{\varepsilon}{\xi} - (T_M^0 - \theta) \cdot \alpha(v), \quad (7)$$

где  $T_M^0$  - температура модуля,  $S(t)$  - енергія сонячної інсоляції [Вт/м<sup>2</sup>],  $\theta$  - температура повітря,  $v$  - швидкість вітру,  $\alpha(v)$  - коефіцієнт тепловіддачі модуля, що враховує при заданій швидкості вітру теплопровідність повітря, його кінематичну в'язкість, коефіцієнт теплового розширення та критерії Прандтля, Релея, Нуссельта,  $m$  - маса модуля, віднесена до одиниці площі його поверхні [кг/м<sup>2</sup>],  $\varepsilon$  - коефіцієнт чорноти модуля,  $\xi$  - коефіцієнт охолодження модуля ( $\xi = 1$ , якщо охолодження з одного боку поверхні модуля,  $\xi = 2$ , якщо охолодження з двох сторін).

Відмінними рисами запропонованого підходу до побудови моделей фотоелектричного перетворення сонячної енергії є:

- 1) факторизація функції фотоелектричного перетворення у вигляді добутку приватних функцій впливу,
- 2) аналітичне обчислення цих функцій за критерієм мінімальної середньоквадратичної помилки прогнозу,

Факторизація функції фотоелектричного перетворення істотно спрощує модель і скорочує час навчання внаслідок виключення з розгляду численних комбінацій, що слабо впливають одночасно діючих вхідних факторів. Аналітичне обчислення функцій впливу шляхом розв'язання систем рівнянь алгебри зменшує час, необхідний для навчання PV-моделей, порівняно з методами послідовного наближення, наприклад, що базуються на використанні нейронних мереж. Цифрова систематизація та усереднення, отриманих прямим обчисленням, функцій впливу створюють умови для



старту PV-моделей за мінімальної глибини масиву ретроспективних даних, що використовуються для навчання моделей. Це особливо важливо для прогнозування роботи нових станцій, що підключаються до енергосистеми.

Основною умовою для використання запропонованої моделі є глибина бази архівних даних, за якими проводиться навчання. Вона повинна бути більшою за довжину будь-якої послідовності, що входять до неї, однакових (не змінюються) даних. В іншому випадку внаслідок сингулярності зворотної системної матриці стають необчислюваними функції впливу, якими визначається прогноз.

Подальший розвиток запропонованого підходу бачиться у вдосконаленні алгоритму обчислення сонячної радіації з урахуванням дифузної та відбитої складових, а також облік тренду помилок метеорологічних прогнозів погоди.

### 3.2 Алгоритм обчислення рівня сонячної інсоляції.

Параметри взаємної орієнтації сонця та фотоелектричного модуля ілюструються на рис.3.1

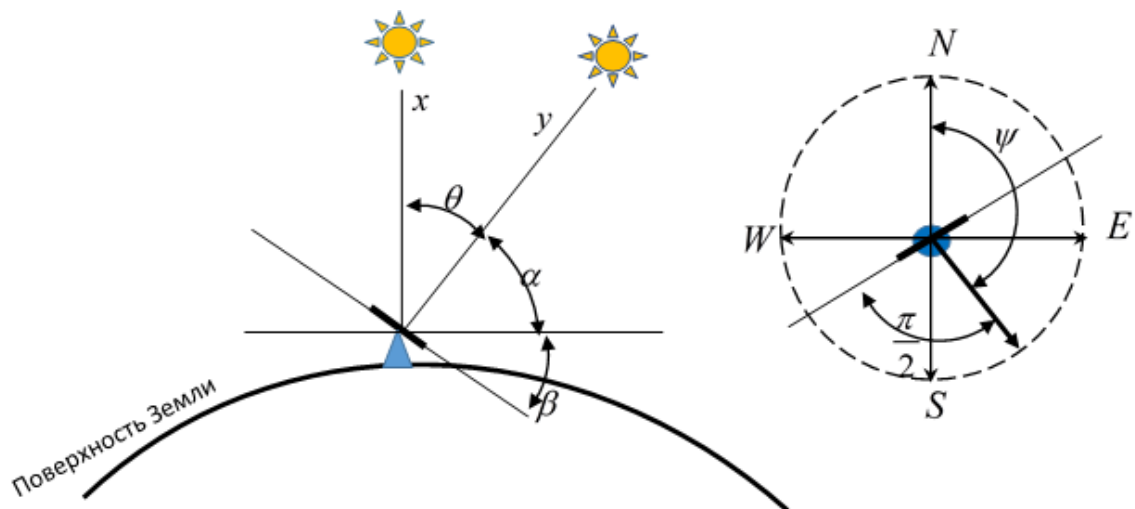


Рис.3.1. Параметри взаємної орієнтації сонця та фотоелектричного модуля

$x$  - найкоротша довжина шляху, що проходить світлом через атмосферу,  $y$  - довжина шляху, що проходить світлом через атмосферу,  $\theta$  - зенітний кут,  $\alpha$  - кут піднесення,  $\beta$  - кут нахилу модуля,  $\psi$  - полярний кут, на який повернутий модуль.

Початкові дані:

$\beta$  - кут нахилу модуля,

$\psi$  - полярний кут, на який повернутий модуль

дані геолокації:

$\varphi$  - широта розташування модуля,

*Longitude* - довгота розташування модуля.

Відмінювання Сонця  $\delta$  - кут між екватором і уявною лінією, що з'єднує центри Землі та Сонця

$$\delta = \arcsin \left\{ \sin(23,45^\circ) \sin \left[ \frac{360}{365} (d - 81) \right] \right\}$$

*Місцевий стандартний часовий меридіан (LSTM)*

$$LSTM = 15^\circ \cdot \Delta T_{GMT},$$

де  $\Delta T_{GMT}$  - різниця між місцевим часом та середнім часом за Гринвічем у годиннику,  $15^\circ = 360^\circ / (24 \text{ години})$ .

Рівняння часу в хвилинах - емпіричне рівняння, що враховує виправлення на ексцентриситет земної орбіти і нахил земної осі.

$$EoT = -9,87 \sin(2B) + 7,53 \cos(B) + 1,5 \sin(B),$$

де  $B = \frac{360}{365}(d - 81)$  в кутах,  $d$  - кількість днів з початку року.

Тимчасовий поправочний коефіцієнт (у хвилинах) розраховується для врахування зміни місцевого сонячного часу в одній часовій зоні з урахуванням довготи в межах цієї зони, а також враховує рівняння часу:

$$TC = 4(-LSTM + Longitude) + EoT$$

Місцевий сонячний час у годиннику

$$LST = LT + \frac{TS}{60},$$

де  $LT$  - місцевий час в годинах.

Годинний кут переводить місцевий сонячний час у кількість градусів, що сонце проходить небом.

$$HRA = 15^{\circ}(LST - 12)$$

Кут піднесення  $\alpha$

$$\alpha = \arcsin[\sin \delta \sin \varphi + \cos \delta \cos \varphi \cos(HRA)],$$

де  $\varphi$  - широта розташування модуля

Виправлення на кут висоти сонця

$$\alpha^c = \alpha + 0,061359 \cdot \frac{0,1594 + 1,123 \cdot \alpha + 0,065656 \cdot \alpha^2}{1 + 28,9344 \cdot \alpha + 277,3971 \cdot \alpha^2}$$

Атмосферна маса АМ – відношення атмосферного шляху, що проходить променем світла до найкоротшого шляху  $\frac{y}{x}$ . З урахуванням кулястості Землі дорівнює:

$$AM = \frac{\exp(-z/8435,2)}{\sin \alpha^c + 0,50572(6,07995 + \alpha^c)^{-1,6364}},$$

де  $z$  – висота над рівнем моря, м, 8435,2 – Релеївська висота атмосфери.

Релеївська оптична товщина атмосфери

$$\delta_R(m) = \begin{cases} (6,6296 + 1,7513 \cdot m - 0,1202 \cdot m^2 + 0,0065 \cdot m^3 - 0,00013 \cdot m^4)^{-1} & \text{for } m \leq 20, \\ (10,4 + 0,718 \cdot m)^{-1} & \text{for } m > 20. \end{cases}$$

Радіація чистого піднебіння на горизонтальну поверхню

$$G_D = 1,367 \cdot \left[ \exp(-0,8662 \cdot T_{LK} \cdot m \cdot \delta_R(m)) \right],$$

Коефіцієнт каламутності Лінке

$$T_{LK} = \begin{cases} 2 & \text{for } \theta \leq -10^0 C, \\ 3 & \text{for } \theta \geq +25^0 C, \\ 2 + \frac{\theta + 10}{35} & \text{for } -10^0 C < \theta < +25^0 C \end{cases}$$

Функція дифузного пропускання є функцією коефіцієнта каламутності Лінке

$$Tn(T_{LK}) = -0,015843 + 0,030543 \cdot T_{LK} + 0,0003797 \cdot T_{LK}^2$$

Функція сонячної висоти

$$F_d(\alpha, T_{LK}) = A_0 + A_1 \sin \alpha + A_2 \sin^2 \alpha$$

$$A_0 = \begin{cases} 0,0022/Tn(T_{LK}) & \text{for } A_0 \cdot Tn(T_{LK}) < 0,0022 \\ 0,26463 - 0,061581 \cdot T_{LK} - 0,011161 \cdot T_{LK}^2 & \text{otherwise} \end{cases}$$

$$A_1 = 2,0402 + 0,018945 \cdot T_{LK} - 0,011161 T_{LK}^2$$

$$A_2 = -1,3025 + 0,039231 \cdot T_{LK} + 0,0085079 T_{LK}^2$$

Згідно моделі *ESRA*, розсіяна освітленість ясного неба

$$G_d = 1,367 \cdot T_n(T_{LK}) \cdot F_d(\alpha)$$

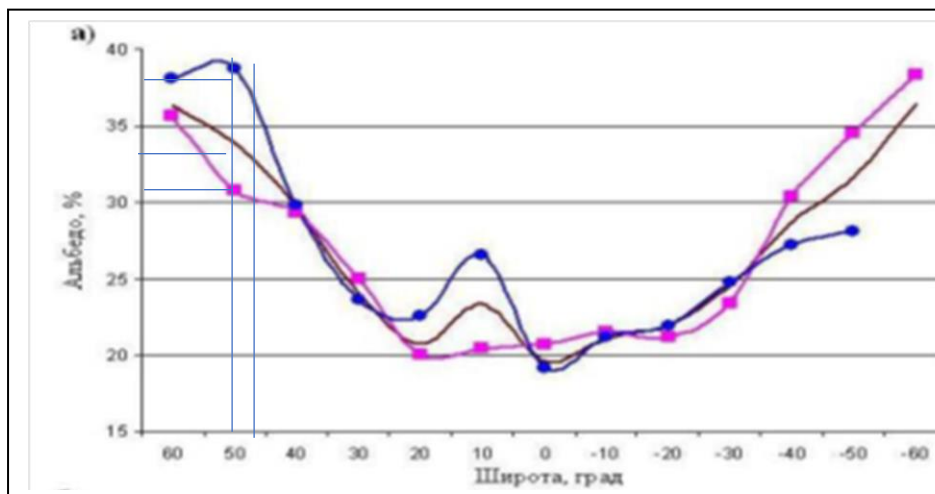
Полярний кут Сонця  $\Theta$  - напрямком компаса убік, звідки рухається Сонце

$$\Theta = \begin{cases} 360 - \arccos \left[ \frac{\sin \delta \cos \varphi - \cos \delta \sin \varphi \cos(HRA)}{\cos \alpha} \right], & \text{if } LST > 12 \text{ or } HRA > 0 \\ \arccos \left[ \frac{\sin \delta \cos \varphi - \cos \delta \sin \varphi \cos(HRA)}{\cos \alpha} \right], & \text{if } LST < 12 \text{ or } HRA < 0 \end{cases}$$

Відбита радіація

$$R_{inc} = \frac{(G_D + G_d) A_k (1 - \cos \beta)}{2},$$

де  $A_k$  — альbedo земної поверхні.



Для широт українських станцій:

Середньорічне альbedo 0,34 (коричнева лінія)

Влітку 0,37 (синя лінія)

Взимку 0,31 (рожева лінія)

Наближена оцінка альbedo до функцій місяця  $m=1 \dots 12$

$$A_k = 0,06 \sin\left(\frac{\pi}{12} m\right) + 0,31$$

*Інтенсивність сонячного випромінювання на поверхню модуля із заданою орієнтацією та кутом нахилу*

$$s(t) = P_{nom} \cdot G_D \cdot [\cos \alpha \sin \beta \cos(\psi - \Theta) + \sin \alpha \cos \beta] + G_d + R_{inc}, \text{ [кВт/кв.м]}$$

### **3.3 Розробка машинно-навченої прогностичної моделі процесу генерації фотоелектричної енергії з урахуванням метеорологічних факторів, що випадково змінюються.**

У машинному навчанні, при аналізі даних, перед розробником ставиться проблема побудови моделі, що пояснює ці дані, вона повинна спростити сприйняття користувачем. Для розробки машинно-навчальної моделі було обрано алгоритм прогнозу методом найменших квадратів

Прогнозовану годинну генерацію  $P(t)$  шукаємо у вигляді

$$P_0(t) = S(t) \cdot F(\tau, \sigma, \theta, v, h, d, m, r) \quad (1)$$

де  $F(\tau, \sigma, \theta, v, h, d, m, r)$  - функція впливу

Навчальний масив даних глибиною 1 рік погодинних значень має такий вигляд:

Порядковий номер години у навчальній послідовності:  $t = 1, 2, 3, \dots, i, \dots, n$ ,

Питома годинна генерація: .....  $P_0(t) = P_0(1), P_0(2), \dots, P_0(i), \dots, P_0(n)$ ,

Альbedo хмар: .....  $\tau(t) = \tau(1), \tau(2), \dots, \tau(i), \dots, \tau(n)$ ,

If (Cloud Total)  $\neq 0$

$$\tau(t) = 0,7(\text{Cloud Low}) / (\text{Cloud Total}) + 0,3(\text{Cloud Medium}) / (\text{Cloud Total})$$

If (Cloud Total) = 0

$$\tau(t) = 0$$

Хмарність: .....  $\sigma(t) = \sigma(1), \sigma(2), \dots, \sigma(i), \dots, \sigma(n)$ ,

Температура повітря: .....  $\theta(t) = \theta(1), \theta(2), \dots, \theta(i), \dots, \theta(n)$ ,

Швидкість вітру: .....  $v(t) = v(1), v(2), \dots, v(i), \dots, v(n)$ .

Години.....  $h(t)$

День .....  $d(t)$

Місяць.....  $m(t)$

*Нормалізація даних.* Усі дані наводяться до нормального вигляду у відносних одиницях.

$$\tau(t) = \frac{\tau^*(t)}{\tau_{\text{sup}}^*}, \quad \sigma(t) = \frac{\sigma^*(t)}{\sigma_{\text{sup}}^*}, \quad \theta(t) = \frac{T^*(t)}{T_{\text{sup}}^*}, \quad v(t) = \frac{V^*(t)}{V_{\text{sup}}^*}, \dots$$

\*- вихідні одиниці виміру

$$\tau_{\text{sup}}^* = 100\%, \quad \sigma_{\text{sup}}^* = 100\%, \quad T_{\text{sup}}^* = 50^{\circ}, \quad V_{\text{sup}}^* = 50 \text{ м/с}$$

Якщо в прогнозі дається діапазон, набуваємо середнього значення.

Обмеження на середні значення:

$$0,2 \leq \tau(t) \leq 1,0,$$

$$0,0 \leq \sigma(t) \leq 0,9,$$

$$0,05 \leq v(t) \leq 0,2.$$

*Функції впливу.* Залежність функції впливу  $F(\tau, \sigma, \theta, v, h, d, m, r)$  від впливових параметрів шукатимемо у вигляді твору

$$F(\tau, \sigma, \theta, v, h, d, m, r) = Tr(\tau) \cdot Cl(\sigma) \cdot \eta_T(\theta) \cdot \eta_w(v) \cdot H(h) \cdot D(d) \cdot M(m)$$

Функція впливу альbedo

$$Tr(\tau) = \exp\left\{-\left(a_0 + a_1\tau + a_2\tau^2 + a_3\tau^3 + a_4\tau^4 + \dots\right)\right\}$$

Функція впливу хмарності

$$Cl(\sigma) = \exp\left\{-\left(b_0 + b_1\sigma + b_2\sigma^2 + b_3\sigma^3 + b_4\sigma^4 + \dots\right)\right\}$$

Функція впливу температури

$$\eta_T(\theta) = \exp\left\{-\left(c_0 + c_1\theta + c_2\theta^2 + c_3\theta^3 + c_4\theta^4 + \dots\right)\right\}$$

Функція впливу вітру

$$\eta_W(v) = \exp\left\{-\left(d_0 + d_1v + d_2v^2 + d_3v^3 + d_4v^4 + \dots\right)\right\}$$

Функція впливу часу протягом доби

$$H(h) = \exp\left\{-\left(f_1h + f_2h^2 + \delta_3h^3 + \delta_4h^4\right)\right\} \quad (6)$$

Функція впливу доби протягом місяця

$$D(d) = \exp\left\{-\left(g_1d + g_2d^2 + g_3d^3 + g_4d^4\right)\right\} \quad (7)$$

Функція впливу місяця протягом року

$$M(m) = \exp\left\{-\left(q_1m + q_2m^2 + q_3m^3 + q_4m^4\right)\right\} \quad (8)$$

Роозділив обидві частини рівняння (1) на  $S(t)$ , отримаємо рівняння другого приближення

$$\frac{P_0(t)}{S(t)} = F(\tau, \sigma, \theta, v, h, d, m, r) \quad (9)$$

Логарифмуючи цей вираз, отримаємо

$$\ln \frac{S(t)}{P(t)} = \mathbf{P}_t[\tau(t), \sigma(t), \theta(t), v(t), H(t), D(t), M(t)] \quad , \quad (10)$$



де поліном  $\mathbf{P}_t$  має вигляд:

$$\begin{aligned} \mathbf{P}_t = & r + (a_1\tau + a_2\tau^2 + a_3\tau^3 + a_4\tau^4 + \dots) + (b_1\sigma + b_2\sigma^2 + b_3\sigma^3 + b_4\sigma^4 + \dots) + \\ & + (c_1\theta + c_2\theta^2 + c_3\theta^3 + c_4\theta^4 + \dots) + (\delta_1v + \delta_2v^2 + \delta_3v^3 + \delta_4v^4 + \dots) + \\ & + (f_1h + f_2h^2 + \delta_3h^3 + \delta_4h^4 + \dots) + (g_1d + g_2d^2 + g_3d^3 + g_4d^4 + \dots) + \\ & + (q_1m + q_2m^2 + q_3m^3 + q_4m^4 + \dots) \end{aligned} \quad (11)$$

Коефіцієнти поліномів впливу знаходимо з умови мінімізації середньоквадратичного відхилення від логарифмічної навчальної послідовності

$$\sum_{t=1}^n \left\{ \ln \frac{S(t)}{P(t)} - \mathbf{P}_t \right\}^2 = \min. \quad (12)$$

Диференціюючи ліву частину (12) за шуканими коефіцієнтами полінома та прирівнюючи приватні похідні нулю, отримуємо систему рівнянь (далі скрізь верхній індекс, взятий у дужки, означає показник ступеня, в який зводиться змінна, верхній індекс без дужок вказує порядковий номер змінної)

$$\sum_{t=1}^n \mathbf{P}_t = \sum_{t=1}^n \ln \frac{S(t)}{P(t)},$$

$$\sum_{t=1}^n \mathbf{P}_t \cdot \tau^{(1)}(t) = \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot \tau^{(1)}(t); \quad \dots \quad \sum_{t=1}^n \mathbf{P}_t \cdot \tau^{(4)}(t) = \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot \tau^{(4)}(t);$$

$$\sum_{t=1}^n \mathbf{P}_t \cdot \sigma^{(1)}(t) = \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot \sigma^{(1)}(t); \quad \dots \quad \sum_{t=1}^n \mathbf{P}_t \cdot \sigma^{(4)}(t) = \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot \sigma^{(4)}(t);$$

$$\sum_{t=1}^n \mathbf{P}_t \cdot \theta^{(1)}(t) = \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot \theta^{(1)}(t); \quad \dots \quad \sum_{t=1}^n \mathbf{P}_t \cdot \theta^{(4)}(t) = \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot \theta^{(4)}(t);$$

$$\sum_{t=1}^n \mathbf{P}_t \cdot v^{(1)}(t) = \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot v^{(1)}(t); \dots \sum_{t=1}^n \mathbf{P}_t \cdot v^{(4)}(t) = \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot v^{(4)}(t);$$

$$\left| \begin{array}{l} \sum_{t=1}^n \mathbf{P}_t \cdot h^{(1)}(t) = \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot h^{(1)}(t); \dots \sum_{t=1}^n \mathbf{P}_t \cdot h^{(4)}(t) = \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot h^{(4)}(t) \\ \sum_{t=1}^n \mathbf{P}_t \cdot d^{(1)}(t) = \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot d^{(1)}(t); \dots \sum_{t=1}^n \mathbf{P}_t \cdot d^{(4)}(t) = \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot d^{(4)}(t) \\ \sum_{t=1}^n \mathbf{P}_t \cdot m^{(1)}(t) = \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot m^{(1)}(t); \dots \sum_{t=1}^n \mathbf{P}_t \cdot m^{(4)}(t) = \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot m^{(4)}(t). \end{array} \right. \quad (13)$$

Для вирішення цієї системи рівнянь наводимо її до канонічного матричного вигляду

$$\begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_i \\ \vdots \\ x_{28} \end{pmatrix} \times \begin{pmatrix} A_0^0 & A_1^0 & \dots & A_i^0 & \dots & A_{28}^0 \\ A_0^1 & A_1^1 & \dots & A_i^1 & \dots & A_{28}^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_0^j & A_1^j & \dots & A_i^j & \dots & A_{28}^j \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_0^{28} & A_1^{28} & \dots & A_i^{28} & \dots & A_{28}^{28} \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_j \\ \vdots \\ y_{28} \end{pmatrix}. \quad (14)$$

Тут зазначено:

$$\left| \begin{array}{l} x_0 = r, x_1 = a_1, x_2 = a_2, x_3 = a_3, x_4 = a_4, x_5 = b_1, x_6 = b_2, x_7 = b_3, x_8 = b_4, \\ x_9 = c_1, x_{10} = c_2, x_{11} = c_3, x_{12} = c_4, x_{13} = \delta_1, x_{14} = \delta_2, x_{15} = \delta_3, x_{16} = \delta_4, \\ x_{17} = f_1, x_{18} = f_2, x_{19} = f_3, x_{20} = f_4, x_{21} = g_1, x_{22} = g_2, x_{23} = g_3, x_{24} = g_4, \\ x_{25} = q_1, x_{26} = q_2, x_{27} = q_3, x_{28} = q_4. \end{array} \right.$$

Праві частини системи рівнянь обчислюються за формулами:

$$\begin{aligned}
y_0 &= \sum_{t=1}^n \ln \frac{S(t)}{P(t)}, \\
y_1 &= \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot \tau^{(1)}(t); \dots\dots\dots y_4 = \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot \tau^{(4)}(t); \\
y_5 &= \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot \sigma^{(1)}(t); \dots\dots\dots y_8 = \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot \sigma^{(4)}(t); \\
y_9 &= \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot \theta^{(1)}(t); \dots\dots\dots y_{12} = \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot \theta^{(4)}(t); \\
y_{13} &= \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot \nu^{(1)}(t); \dots\dots\dots y_{16} = \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot \nu^{(4)}(t); \\
y_{17} &= \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot h^{(1)}(t); \dots\dots\dots y_{20} = \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot h^{(4)}(t); \\
y_{21} &= \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot d^{(1)}(t); \dots\dots\dots y_{24} = \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot d^{(4)}(t); \\
y_{25} &= \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot m^{(1)}(t); \dots\dots\dots y_{28} = \sum_{t=1}^n \ln \frac{S(t)}{P(t)} \cdot m^{(4)}(t). \quad (16)
\end{aligned}$$

Елементи матриці А обчислюються ітеративним способом за формулами:

$$A_i^j = \sum_{t=1}^n \alpha_i^j(t), \quad i = 0...28; j = 0...28; \quad (17)$$

$$\alpha_i^0 = \begin{cases} [\tau(t)]^{(i)}, & i = 0...4; \\ [\sigma(t)]^{(i-4)}, & i = 5...8; \\ [\theta(t)]^{(i-8)}, & i = 9...12; \\ [v(t)]^{(i-12)}, & i = 13...16; \\ [h(t)]^{(i-16)}, & i = 17...20; \\ [d(t)]^{(i-20)}, & i = 21...24; \\ [m(t)]^{(i-24)}, & i = 25...28; \end{cases} \quad (18)$$

$$\alpha_i^j(t) = \begin{cases} \alpha_i^{j-1}(t) \cdot \tau(t), & i = 0...28, j = 1...4; \\ \alpha_i^{j-5}(t) \cdot \sigma(t), & i = 0...28, j = 5...8; \\ \alpha_i^{j-9}(t) \cdot \theta(t), & i = 0...28, j = 9...12; \\ \alpha_i^{j-13}(t) \cdot \delta(t), & i = 0...28, j = 13...16; \\ \alpha_i^{j-17}(t) \cdot h(t), & i = 0...28, j = 17...20; \\ \alpha_i^{j-21}(t) \cdot d(t), & i = 0...28, j = 21...24; \\ \alpha_i^{j-25}(t) \cdot m(t), & i = 0...28, j = 25...28; \end{cases} \quad (19)$$

Наприклад, другий елемент другої строки матриці А

$$\alpha_1^1(t) = \alpha_1^0(t) \tau(t) = \tau(t) \cdot \tau(t),$$

$$A_1^1 = \sum_{t=1}^n \alpha_1^1(t) = \sum_{t=1}^n \tau^{(2)}(t).$$

Наприклад, п'ятий елемент другої строки матриці А

$$\alpha_5^1(t) = \alpha_5^0(t) \tau(t) = \sigma(t) \cdot \tau(t),$$

$$A_5^1 = \sum_{t=1}^n \alpha_5^1(t) = \sum_{t=1}^n \sigma(t) \cdot \tau(t).$$

Рішення системи рівнянь (14) дає значення коефіцієнтів поліномів, якими визначаються функції впливу.

Довірчий прогноз (видалення неправдоподібних даних з навчальної послідовності) Исходные данные:

- 1) Прогнозована генерація  $P(t)$  (forecast) на навчальній послідовності
- 2) Виміряна генерація  $P_0(t)$  на навчальній послідовності

## 1. Алгоритм

- 1) Модуль абсолютної помилки

$$\Delta(t) = |P(t) - P_0(t)|$$

- 2) Оцінка середнього значення абсолютної помилки на всьому періоді навчальної послідовності

$$\bar{\Delta} = \frac{\sum_t \Delta(t)}{N}$$

- 3) Оцінка середньоквадратичного відхилення помилки

$$\sigma = \sqrt{\frac{\sum_t [\bar{\Delta} - \Delta(t)]^2}{N(N-1)}}$$

$N$  - кількість годин, що враховуються, на всьому періоді навчальної послідовності (кількість доданків у сумах).

4) Довірчий інтервал помилки

$$s = CF \cdot \sigma$$

$CF$  - коефіцієнт довіри (Confidence Factor)

5) Умова неправдоподібного прогнозу

$$\Delta(t) > s + \bar{\Delta} \quad \Delta(t) > CF \cdot \bar{\Delta}$$

6) Знаходимо кількість неправдоподібних прогнозів  $\nu$ .

Якщо ця кількість менша або дорівнює  $FD \cdot N$ , виключаємо їх усіх із навчальної послідовності та шукаємо друге наближення  $P'(t)$

(forecast-2), використовуючи лише «правдоподібні» дані, що

залишилися..  $FD$ - коефіцієнт видалення неправдивих даних (False Data Removal Factor) Якщо кількість неправдоподібних даних

більша, ніж  $FD \cdot N$ , сортуємо всі неправдоподібні помилки щодо

спадання і виключаємо  $\nu' \leq 0,1N$ , більших помилок. Шукаємо друге

наближення  $P'(t)$ , використовуючи нову навчальну послідовність з

кількістю даних  $N' = N - \nu'$

7) Повторюємо процедуру виключення. Шукаємо (forecast-3) и т.д. ....

Критерії зупинки:

- а) з навчальної послідовності виключено  $FD \cdot N$  вихідних даних (годин), або
- б) відсутні неправдоподібні значення  $\forall(t) [\Delta(t) \leq s + \bar{\Delta}]$ .

### 3.4 Розробка та дослідження моделі поведінки фотоелектричного модуля у критичних термічних умовах.

Тепло, одержуване пластиною, витрачається на нагрівання пластини і втрати в довкілля.

У цьому втрати тим більше, що стоїть температура пластини.

При сталості потоку тепла, що підводиться, у міру нагрівання пластини

втрати ростуть, а нагрівання сповільнюється. В якийсь момент втрати стають рівні кількості тепла, що підводиться, нагрівання припиняється, температура стабілізується.

Досягається рівновага:

$$Q_c = \alpha \cdot F \cdot (t_{пл} - t_v)$$

Де:

- $Q_c$  – тепло, що підводиться в одиницю часу, ккал/ч або кВт;
- $F$  – поверхня пластини,  $m^2$ ;
- $t_{пл}$  – температура пластини,  $^{\circ}C$  або  $K$ ;
- $t_v$  – температура повітря,  $^{\circ}C$  або  $K$ ;
- $\alpha$  – коефіцієнт тепловіддачі від пластини до повітря, що залежить від безлічі факторів: температура пластини, матеріал пластини та ін.
- Проте найбільше впливає інтенсивність тепловіддачі надає швидкість вітру. Нехтуючи рештою факторів, можна прийняти:

Швидкість вітру, м/с	$\alpha$ , Вт/м <sup>2</sup> К
0,6	7,5
0,9	9
1,2	12
1,5	15

1,8	17,5
2,1	15
2,4	16
2,7	17,5
3,0	18,7
3,3	19
4,2	22,5
5,1	25
6	27,5
6,9	30
7,8	32
8,7	34
9,6	36
10,5	38
11,4	40
12,3	41,7
13,2	43,5
14,1	45
15	47

### Модель

#### 1. Параметри модуля

Масса .....  $m = 18$  [кг]

Площа поверхності.....  $F = 1,42$  [м<sup>2</sup>]

Теплоємність.....  $C = 900$  [Дж/кг/К<sup>0</sup>]

Температурний коефіцієнт потужності .....  $k_p = (0,4) 4,0$  [%/град].

Термічний поріг зниження потужності .....  $\Theta_p = (25) 49$  [С<sup>0</sup>]



Коефіцієнт поглинання теплової енергії ..... $h = 0,4$

## 2. Рівняння теплового балансу

$$cm \frac{d\theta_p}{dt} = S \cdot F \cdot h - (\theta_p - \theta_a) \cdot \alpha(V) \cdot F,$$

де  $\theta_p$  - усереднена на годині температура модуля,  $S$  - усереднена на годині енергія сонячної інсоляції [Вт/м<sup>2</sup>],  $\theta_a$  - усереднена на годині температура повітря,  $\alpha$  - коефіцієнт тепловіддачі модуля (див. табл.),  $V$  - усереднена на годині швидкість вітру.

## 3. Постійна часу нагріву модуля

$$T = \frac{cm}{\alpha F} = \frac{900 \cdot 18}{20 \cdot 1,42} = 570c < \frac{1}{6} \text{ часа}$$

При усередненні на годині можна вважати, що модуль знаходиться в тепловій рівновазі.

## 4. Корекція сонячної інсоляції (для кожної години навчальної послідовності та прогнозованого дня)

Температура модуля

$$\theta_p = \frac{S(t) \cdot \varepsilon}{\alpha(V) \xi} + \theta_a,$$

$\varepsilon = 0,9$  – коефіцієнт чорноти модуля,  $\xi = 2$  – тепловіддача із двох сторін.

Якщо температура модуля перевищує термічний поріг зниження потужності

$\theta_p > \Theta_p$ , то

$$P(t) = P(t) \left[ 1 - \frac{k_p}{100} (\theta - \Theta_p) \right],$$

інакше  $S(t) = S(t)$ .

$$k_p = 0,45\% / C^0$$

$$S_{nom} = 1000 \text{ Вт/м}^2$$

$$\Theta_p = 25 \text{ C}^0$$

В даному розділі були представлені всі необхідні обчислення та вхідні дані для концепції побудовання математичної моделі процесу генерації сонячної енергії. Був налагоджений Алгоритм обчислення рівня сонячної інсоляції. Виконана розробка машинно-навченої прогностичної моделі процесу генерації фотоелектричної енергії з урахуванням метеорологічних факторів, що випадково змінюються, а також розробка та дослідження моделі поведінки фотоелектричного модуля у критичних термічних умовах.

## **Розділ 4**

При побудові системи автоматизації з використанням контролерів з'являється можливість створення системи віддаленого моніторингу та дистанційного керування - найчастіше на базі персонального комп'ютера або пристрою, що його замінює. Такі системи одержали назву систем диспетчеризації.

### **4.1 Принцип роботи**

Сонячні панелі поглинають енергію світла та перетворюють її в електричну енергію, а потім розподіляють електричну енергію через контролер, надають її для безпосереднього використання шляхом моніторингу та зберігають у акумуляторі, а також забезпечують тривале живлення для моніторингу. Системи моніторингу сонячної енергії вимірюють інформацію про виробництво енергії вашою системою, зазвичай від сонячних інверторів або контролера заряду. Постійний струм перетворюється від сонячної батареї в енергію змінного струму для використання у вашому домі. Є багато способів, за допомогою яких можна

отримати доступ до цих даних і зберегти їх; деякі системи пропонують підключення до Інтернету через Wi-Fi та/або Ethernet, а деякі включають функції стільникового зв'язку, щоб ви могли отримати доступ до моніторингу системи під час перебоїв у роботі Інтернету.

1. Коли сонце виділяє велику кількість світла, контролер ділить електричну енергію, перетворену сонячною панеллю, на ту, що використовується для моніторингу, і ту, що зберігається акумулятором;
2. Коли сонце надає малу кількість світла, контролер зніме електричну енергію з батареї та надасть її для моніторингу, коли електричної енергії, перетвореної сонячною енергією, недостатньо для моніторингу нормального використання;
3. Коли сонячної енергії недостатньо, контролер зніме електричну енергію, що зберігається в акумуляторі, для продовження моніторингу.

#### **4.1.1 Мережеві системи моніторингу**

Системи моніторингу сонячних панелей, що підключаються до мережі. Метою мережевої сонячної системи є генерація сонячної енергії для зберігання в комунальній мережі. Це дає вам доступ до енергії, коли це необхідно. Сонячна енергосистема пропонує ряд переваг, якщо є доступ до комунальної мережі. Він має нижчу початкову вартість, оскільки немає необхідності купувати батареї для зберігання виробленої енергії. Моніторинг мережевого зв'язку допомагає підтримувати ефективну роботу всієї сонячної системи, передаючи дані для моніторингу та регулювання виходу сонячної панелі.

#### **4.1.2 Автономні системи сонячного моніторингу**

У автономній системі важливо оптимізувати термін служби батареї, щоб система могла працювати безперебійно та ефективно. Незалежна від мережі

система моніторингу сонячної енергії вимірює рівень заряду батареї, щоб знати, коли настав час заряджатися або коли зарядка закінчилася.

#### **4.1.3 Перевірка продуктивності**

Система може фінансуватися на основі її результатів, і тому користувачеві необхідно виміряти результат і порівняти його з втратами системи. Також може статися так, що виробництво електроенергії потрібно виміряти в замовленні на продаж, або для оцінки деяких «зелених сертифікатів». Трудомісткість і витратність такого обліку визначається кількістю і точністю вимірювань, які необхідно зробити. Найпростіша система, «Розумний» лічильник дозволяє надавати користувачеві зрозумілий зворотний зв'язок у режимі реального часу, щоб перевірити, чи система працює правильно, під час процесу вимірювання виробництва енергії для оплати тарифу тощо.

#### **4.1.4 Оцінка системи**

Для незвичайної системи може бути доцільним виміряти детально функціонування системи, щоб зрозуміти функціонування її компонентів в деталях. Це часто виконувалося в 1980-х і 1990-х, коли системи часто фінансувалися грантами, і головною метою побудови системи було покращити знання та розуміння. Робота фотоелектричних систем тепер добре задокументована, а витрати на моніторинг та оцінку даних зазвичай не можуть бути переоцінені.

#### **4.2 Функціональні можливості АСКОЕ:**

АСКОЕ включає в себе наступні функціональні можливості:

- можливість погодинної передачі даних комерційного обліку до АСКОЕ Головного оператора за допомогою уніфікованого протоколу передачі даних (УППДв) вимірювань, згідно з наданим Головним оператором закодованим Реєстром даних;

- передачі даних до інформаційно-обчислювального комплексу (далі ІОК) Головного оператора через мережу Internet за допомогою технології VPN з використанням протоколів IPSec в режимі Site to Site Tunnel. Шифрування пакетів має здійснюватися за стандартом AES з довжиною ключа 256 біт. Наявність двох вільних інтерфейсів у мережі обладнання суб'єкта ОРЕ, можливість організації динамічної маршрутизації між мережею передачі даних Головного оператора та інформаційною мережею, до якої підключено АСКОЕ суб'єкта ОРЕ, ручного вводу даних по кожному каналу обліку електроенергії.
- Надання до ДП "НЕК "Укренерго" даних комерційного обліку для формування макетів 30817;
- можливість синхронізації сервера АСКОЕ, приладів та пристроїв обліку з джерелом точного часу Інституту метрології та з серверами точного часу АСКОЕ ОРЕ за протоколом NTP відповідно до «Загальних технічних вимог до автоматизованої системи комерційного обліку Оптового ринку електричної енергії України. «Система точного часу та підсистема забезпечення синхронності вимірювання Автоматизованої системи комерційного обліку Оптового ринку електричної енергії України»;
- можливість зберігання в базі даних АСКОЕ первинних та розрахункових даних за обліковими періодами (доба, місяць, квартал, рік) починаючи з дискретності 30 хвилин і вище, протягом терміну позовної давності;
- забезпечення наявності позначок якості інформації (ручний чи автоматизований ввід даних тощо), які повинні формуватися, починаючи з рівня обліку і передаватися на вищі рівні;

- забезпечити наявність резервного серверного обладнання та програмного забезпечення для забезпечення відсутності затримок в передачі даних до ІОК Головного оператора у разі виходу з ладу основного серверного обладнання та програмного забезпечення АСКОЕ суб'єкта ОРЕ;
- Забезпечення резервного живлення лічильників електричної енергії.

Перед введенням в промислову експлуатацію АСКОЕ проходить метрологічну атестацію. В процесі експлуатації АСКОЕ піддається періодичній, позачерговій та іншим видам перевірок, обумовленими ГОСТ 8.513-84 та Закону України «Про метрологію та метрологічну діяльність».

## (АСДУ- автоматизована система диспетчерського управління )

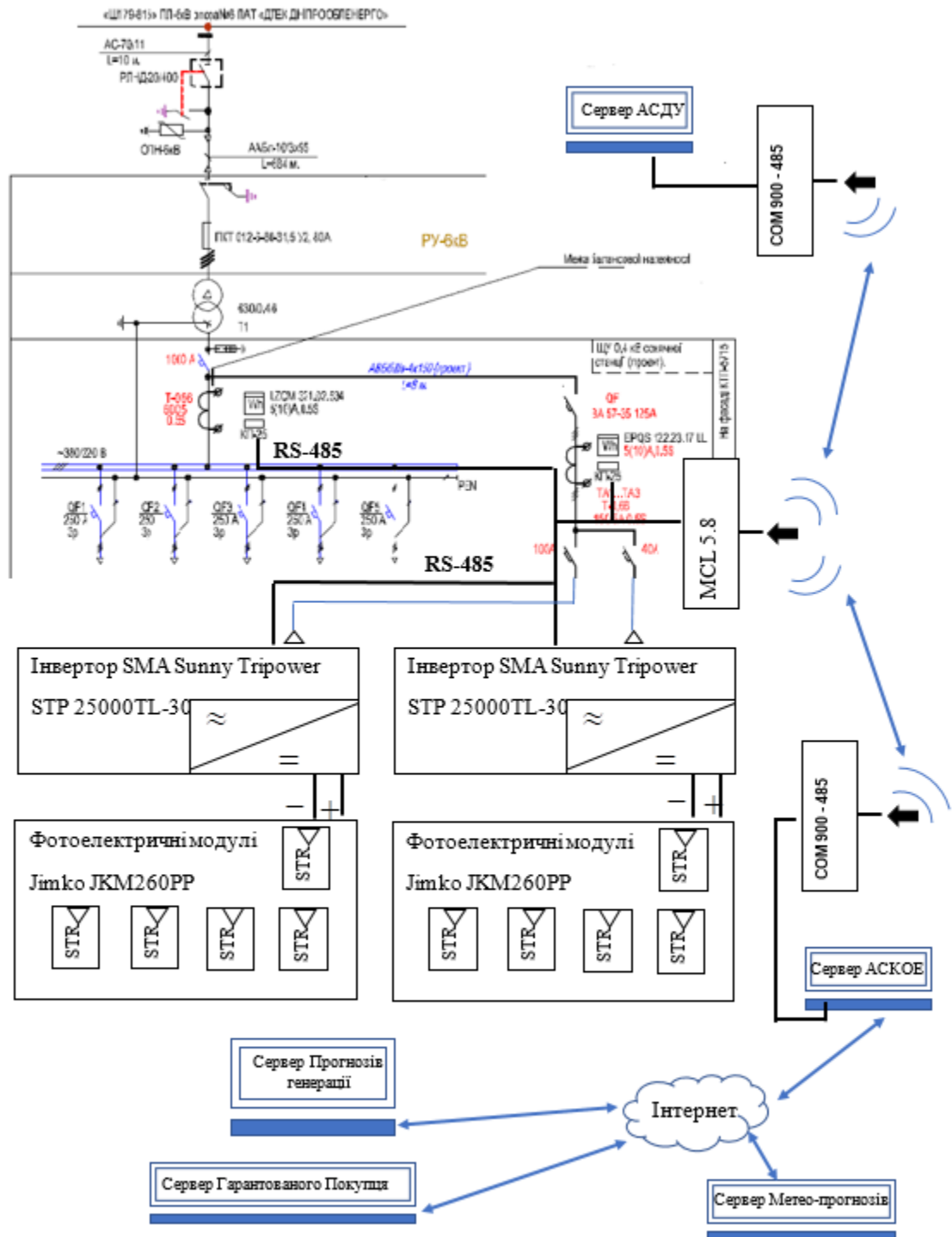


Рис. Комп'ютерна система моніторингу та прогнозування виробітку електроенергії фотогоальванічною станцією. Схема функціональна.

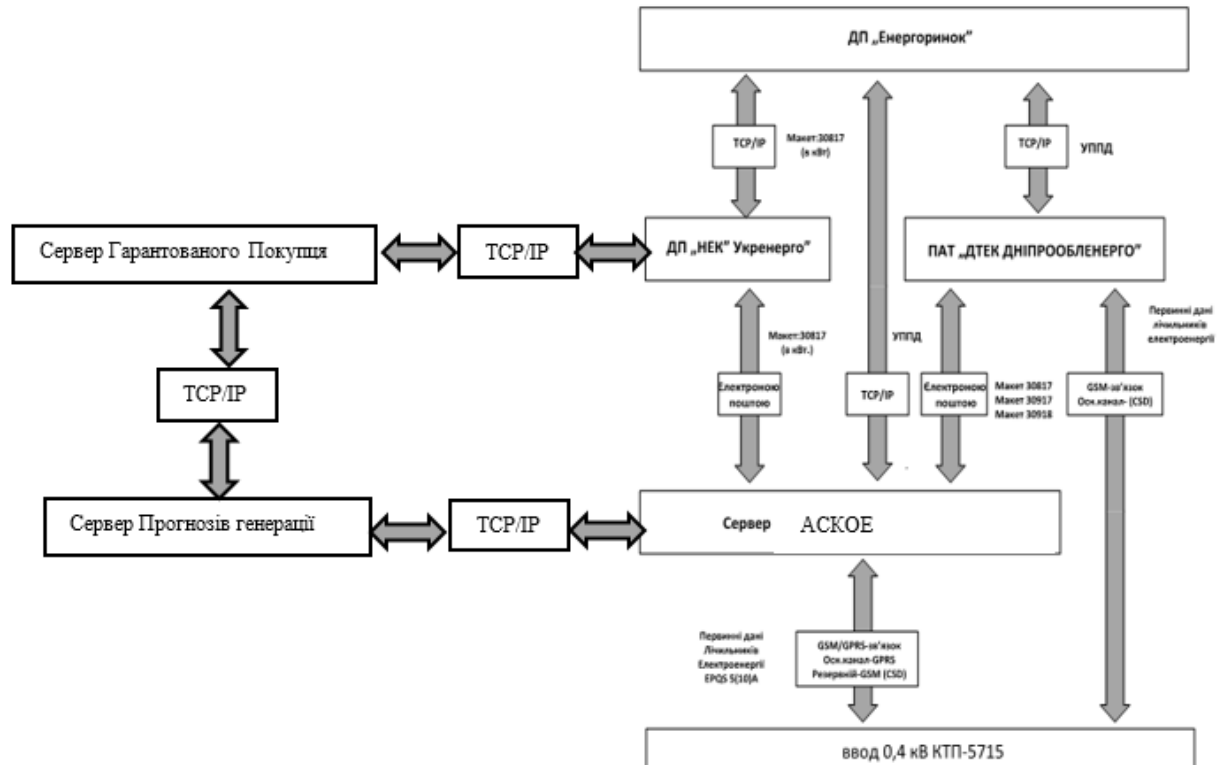


Рис.4.2 Комп'ютерна система моніторингу та прогнозування виробітку електроенергії фотогальванічною станцією. Схема структурна

#### 4.3 Інвертор Струму SUNNY TRIPOWER 15000TL/20000TL/ 25000TL

##### 4.3.1 Опис виробу

Sunny Tripower є безтрансформаторним фотогальванічним інвертором з 2 MPPT-трекерами, який перетворює постійний струм фотогальванічного генератора в трифазний змінний струм, сумісний з мережею, і подає змінний струм в громадську мережу електропостачання.



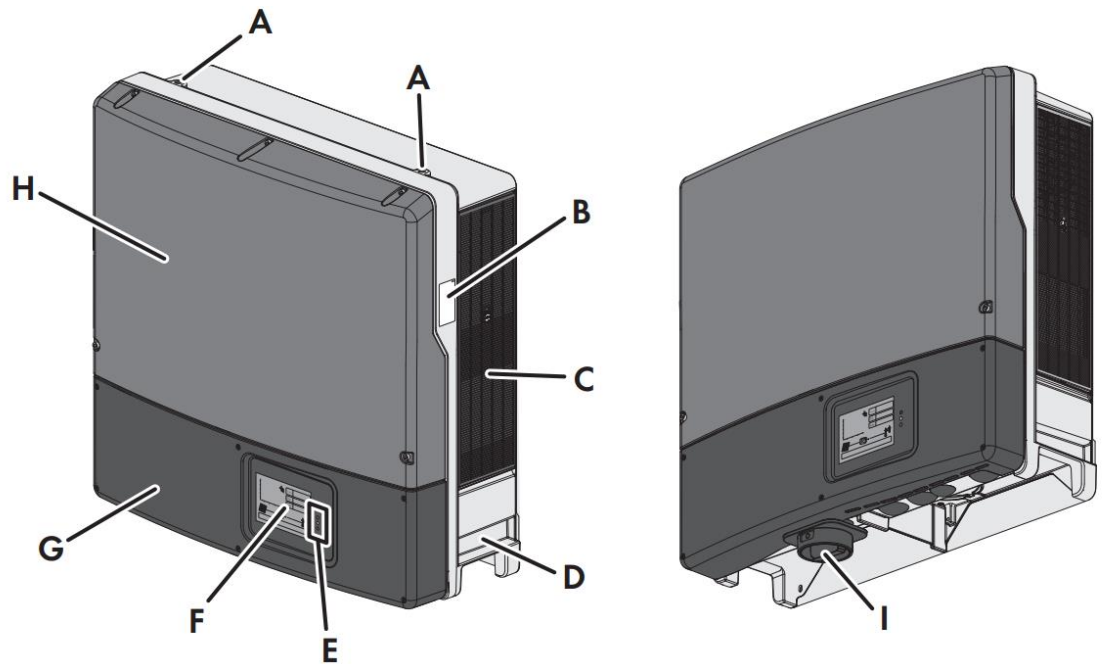


Рис.4.3 Будова Інвертора струму

A - Різьблення для вкручування 2 відкидних болтів для транспортування

B – Заводська табличка. Заводська табличка слугує для ідентифікації інвертора. Відомості на заводській табличці потрібні для безпечного поводження з пристроєм. На заводській табличці наводиться така інформація:

- Тип пристрою (Model)
- Серійний номер (Serial No.)
- Дата виробництва (Date of manufacture)
- Характеристики конкретного пристрою

C Вентиляційні ґрати

D - Втоплена ручка

E – Світлодіоди. Світлодіоди сигналізують про робочий стан інвертора

F - Дисплей (опційно) На дисплеї відображаються поточні експлуатаційні дані та події або помилки.

G - Нижня кришка корпусу

H - Верхня кришка корпусу

I - Вимикач постійного струму Інвертор оснащений вимикачем постійного струму. У положенні I вимикач навантаження постійного струму встановлює з'єднання, що проводить, між фотогальванічним генератором і інвертором. Шляхом перемикачання вимикача навантаження постійного струму в положення O - переривається електричний ланцюг постійного струму, і фотогальванічний генератор повністю відокремлюється від інвертора. Відділення здійснюється за всіма полюсами.

#### **4.3.2 Інтерфейси та функції:**

##### **SMA Speedwire/Webconnect**

Інвертор за замовчуванням оснащений технологією **SMA Speedwire/Webconnect**.

SMA Speedwire/Webconnect є видом підключення на основі стандарту Ethernet. При цьому забезпечується оптимальна для інвертора передача даних зі швидкістю 10/100 Мбіт/с між пристроями Speedwire у фотогальванічних установках та програмним забезпеченням Sunny Explorer. Функція Webconnect забезпечує пряму передачу даних між інверторами невеликих установок та інтернет-порталом Sunny Portal без додаткового пристрою зв'язку, максимальна кількість інверторів на кожну установку з підключенням до Sunny Portal: У великих установках передача даних між інверторами та інтернет-порталом Sunny Portal здійснюється через контролер SMA Cluster Controller. Доступ до інсталяції Sunny Portal можна отримати з будь-якого комп'ютера з підключенням до Інтернету. Для фотогальванічних

установок, що експлуатуються в Італії, Webconnect дозволяє підключати або від'єднувати інвертор від суспільної мережі електропостачання та визначати межі частоти, що застосовуються, за допомогою повідомлень IEC61850-GOOSE.

### **Modbus**

Виріб обладнано інтерфейсом Modbus. Інтерфейс Modbus у стандартному виконанні деактивований і при необхідності має бути налаштований. Інтерфейс Modbus виробів з підтримкою SMA розрахований для промислового застосування, наприклад, у системах SCADA, має наступні завдання:

- дистанційне опитування вимірних значень;
- дистанційне налаштування робочих параметрів;
- встановлення заданих значень для керування установкою.

### **Інтерфейс RS485**

Через інтерфейс RS485 інвертор може за допомогою проводового з'єднання зв'язуватися зі спеціальними пристроями зв'язку SMA (інформацію про продукти SMA, що підтримуються, див. на [www.SMA-Solar.com](http://www.SMA-Solar.com)). Інтерфейс RS485 можна дообладнати та використовувати замість інтерфейсу SMA Speedwire/Webconnect в інверторі.

### **Послуги мережі електропостачання**

Виріб оснащений функціями, що дозволяють надавати послуги мережі електропостачання. Залежно від вимог оператора мережі, ці функції (наприклад, обмеження активної потужності) можна активувати та конфігурувати за допомогою робочих параметрів.

### **SMA Power Control Module**

Модуль SMA Power Control Module дозволяє інвертору продавати послуги мережі електропостачання. Крім того, він додатково

оснащений функціональним реле. SMA Power Control Module). SMA Power Control Module можливо дообладнати.

### **Багатофункціональне реле**

Можна конфігурувати багатофункціональне реле для різних режимів роботи. Багатофункціональне реле призначене, наприклад, для включення та вимикання сигналізаторів пошкоджень (інформацію щодо встановлення та конфігурації див. у посібнику з встановлення багатофункціонального реле). Багатофункціональне реле можна дообладнати.

### **SMA ShadeFix**

Інвертор обладнаний системою SMA ShadeFix для відстеження ефективності залежно від затінення. Система SMA ShadeFix використовує інтелектуальний MPP-трекінг, щоб при затінку знайти робочу точку з найбільшою потужністю. Завдяки системі SMA ShadeFix інвертор завжди використовує найпотужніше джерело живлення фотогальванічних модулів, щоб збільшити вироблення на затінених установках. Система SMA ShadeFix за замовчуванням активована. Часовий інтервал SMA ShadeFix за промовчанням становить 6 хвилин. Це означає, що інвертор шукає оптимальну робочу точку кожні 6 хвилин. Залежно від установки та ситуації затінювання, може знадобитися налаштування часового інтервалу.

Розрядник захисту від перенапруги, тип II

Розрядники захисту від перенапруги обмежують небезпечні перенапруги. Розрядники захисту від перенапруги, тип II можуть бути дообладнані.

### **Q on Demand 24/7**

За допомогою Q on Demand 24/7 інвертор може цілодобово надавати реактивну потужність по всьому ланцюгу пристрою.

## Integrated Plant Control

За допомогою Integrated Plant Control інвертор може відобразити задану оператором мережі графічну характеристику  $Q(U)$  без вимірювання точки підключення до мережі. Устаткування, підключене між інвертором та точкою підключення до мережі, може автоматично компенсуватися інвертором після активації функції.

## SMA Smart Connected

SMA Smart Connected є функцією безкоштовного моніторингу виробу через портал Sunny Portal. За допомогою SMA Smart Connected експлуатуюча сторона та кваліфікований фахівець автоматично та своєчасно отримують інформацію про події, пов'язані з виробом. Функція SMA Smart Connected активується під час реєстрації на порталі Sunny Portal. Для використання SMA Smart Connected виріб повинен бути постійно підключеним до Sunny Portal, і на порталі Sunny Portal повинні бути збережені актуальні дані експлуатуючої сторони та кваліфікованого спеціаліста.

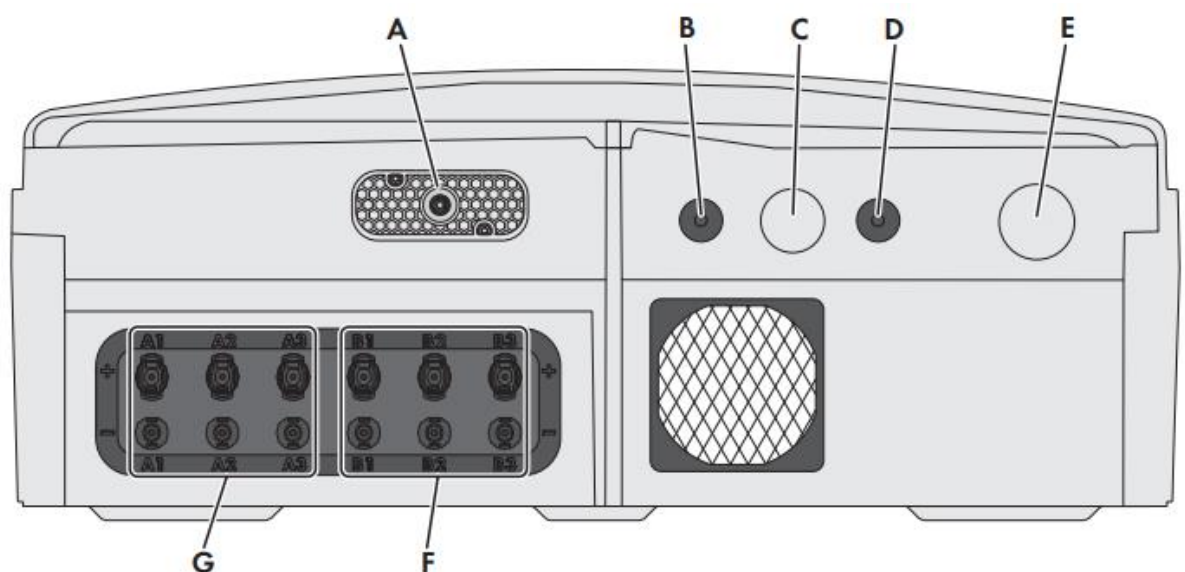


Рис.4.4 Вид знизу

A - Гніздо для вимикача постійного струму

B - Отвір у корпусі M20 із заглушкою для з'єднувального кабелю багатофункціонального реле або SMA Power Control Module

C - Отвір у корпусі M32 із заглушкою для кабелю передачі даних або мережевого кабелю

D - Отвір у корпусі M20 із заглушкою для кабелю передачі даних або мережевого кабелю

E - Отвір у корпусі для підключення змінного струму

F - Позитивні та негативні штепсельні роз'єми постійного струму,  
вхід B

G - Позитивні та негативні штепсельні роз'єми постійного струму,  
вхід A

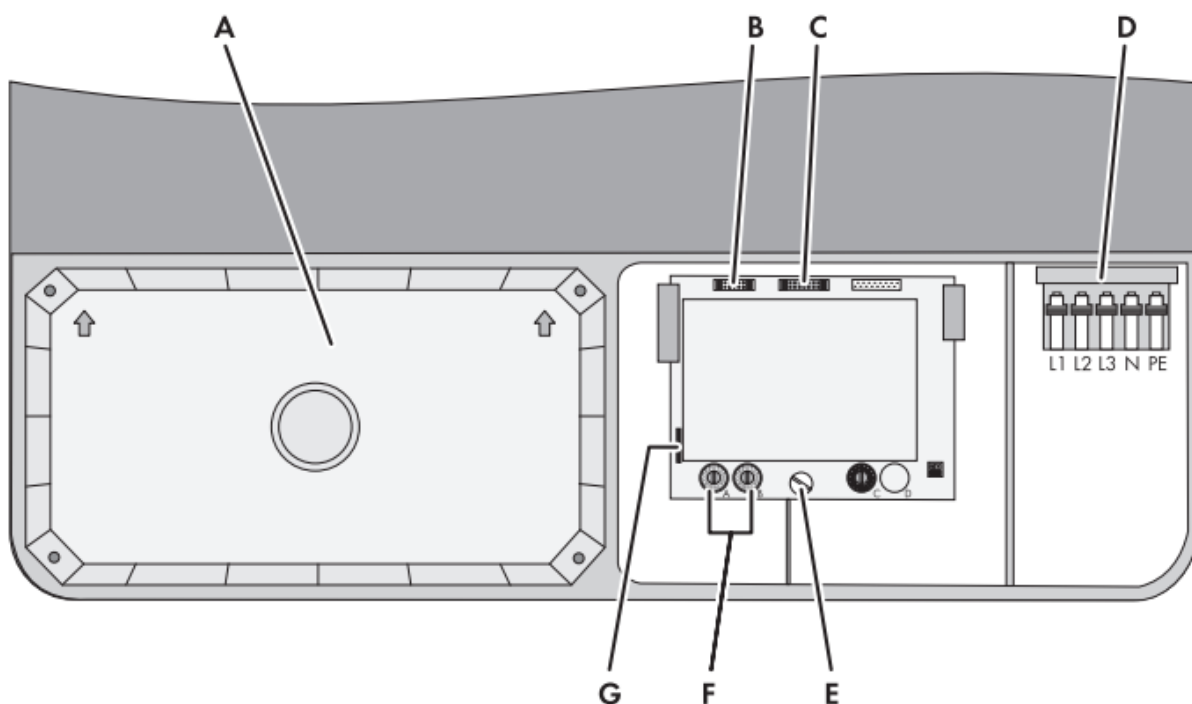


Рис.4.5 Вид зсередини

A - Захисний кожух постійного струму

B - Гніздо для підключення багатофункціонального реле або SMA Power Control Module

C - Гніздо для підключення інтерфейсу зв'язку

D - Притискна планка для підключення кабелю змінного струму

E - Гвинт для від'єднання та закріплення плати зв'язку

F - Поворотні вимикачі A та B для налаштування набору даних країни

G - Гніздо для SD-картки (для сервісних операцій)

### 4.3.3 Технічні характеристики

Вхід постійного струму

Максимальна потужність постійного струму при $\cos \varphi = 1$	25550 Вт
Максимальна вхідна напруга	1000 В
Діапазон напруги МРР	От 390 до 800 В
Номінальна вхідна напруга	600 В
Мінімальна вхідна напруга	150 В
Пускова вхідна напруга	188 В
Максимальний вхідний струм, вхід А	33 А
Максимальний вхідний струм, вхід В	33 А
Максимальний струм короткого замикання на один ряд*	43 А
Максимальний зворотний струм в установці протягом максимум 1 с	0 А
Кількість рядів для кожного входу МРР	3
Категорія перенапруги згідно з ІЕС 62109-1	II

#### Вихід перемінного струму

Розрахункова потужність при 230 В, 50 Гц 25000 Вт	25000 Вт
Максимальна повна потужність змінного струму 25000 ВА	25000 ВА
Розрахункова напруга мережі 230 В	230 В
Номінальна напруга змінного струму 220/230/240 В	220 В/230 В/240 В
Діапазон напруги змінного струму	Від 180 до 280 В



Від 180 до 280 В	
Номінальний змінний струм при 230 В 36,2 А	36,2 А
Максимальний вихідний струм 36,2 А	36,2 А
Максимальний вихідний струм у разі помилки 50 А	50 А
Розрахункова потужність при 230 В, 50 Гц 25000 Вт	25000 Вт
Максимальна повна потужність змінного струму 25000 ВА	25000 ВА
Розрахункова напруга мережі 230 В	230 В
Номінальна напруга змінного струму 220/230/240 В	220 В/230 В/240 В
Сумарні гармонічні спотворення вихідного струму при сумарних гармонічних спотвореннях напруги змінного струму 50 % номінальної потужності	$\leq 3$ %
Номінальна частота мережі	50 Гц
Частота мережі змінного струму*	50 Гц/60 Гц
Робочий діапазон при частоті змінного струму 50 Гц	Від 44 до 55 Гц
Робочий діапазон при частоті змінного струму 60 Гц	Від 54 до 65 Гц
Коефіцієнт потужності за номінальної потужності	1
Коефіцієнт зсуву, регульований	0Перезбудження до

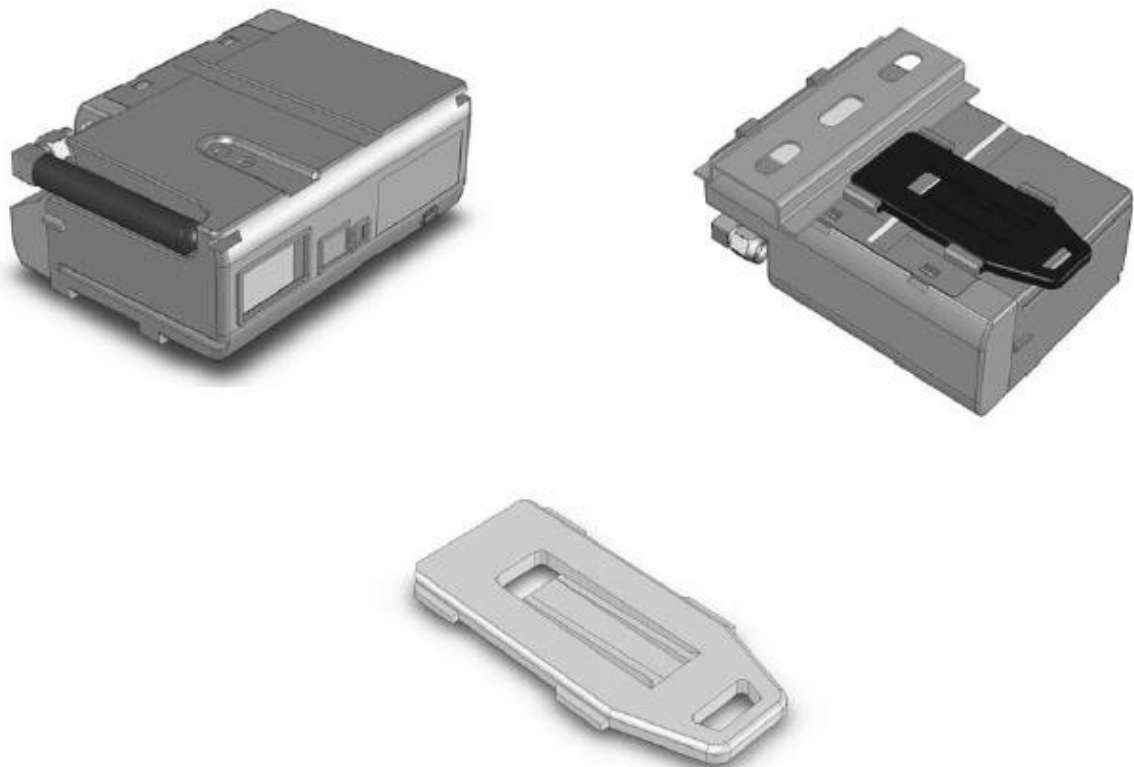
	0Недозбудження
Фази подачі електроенергії до мережі	3
Фази підключення	3
Категорія перенапруги згідно з ІЕС 62109-1	III

#### **4.4 Контролер MCL 5.8 із внутрішнім 4-частотним GSM/GPRS модемом**

##### **4.4.1 Функціональне призначення та застосування контролера MCL 5.8**

Контролер MCL 5.8 (далі в тексті контролер) використовується для автоматизованої передачі даних з приладів обліку електроенергії, тепла та інших енергетичних ресурсів у віддаленому центрі збору даних. Для передачі даних у віддалені центри використовуються CSD/GPRS технології GSM мережі, прозорі (Transparent) або TCP/IP протоколи. Контролер підтримує двосторонній зв'язок для обміну даними (для зчитування даних та параметрування), використовуючи ІЕС 62056-31,

DLMS або IEC 62056-21 (фіксований зв'язок від 300...9600 бод, 8N1 або 7E1 байти) протоколи зв'язку. Для зчитування даних використовуються 20mA



„струмова петля“ та/або RS485 роз'єми електронного зв'язку. Прилад може бути налаштований локально (використовуючи роз'єм RS232) або дистанційно. Контролер використовується для обміну даними в системах з архітектурою „точка-точка“. Виробляються кілька модифікацій контролера з різними можливостями. Контролер може мати внутрішній літійово-полімерний акумулятор або додатковий роз'єм зв'язку для приєднання до місцевої автоматизованої системи зчитування даних (ADNS). Контролер (ADN може бути встановлений під кришкою, що закриває контакти 3-фазного лічильника електроенергії, що виробляється „Elgamaelektronika“, або на 35 мм DIN рейці.(ADNS).

Рис.4.6

#### 4.4.2 Основні модулі контролера

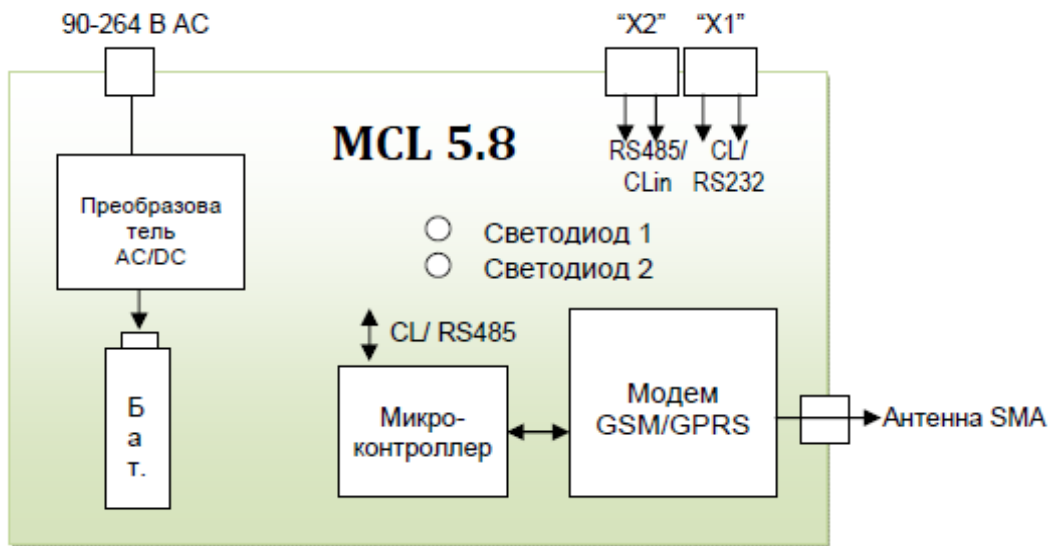


Рис.4.7 Спрощена блок-схема MCL 5.8

До лічильника контролер підключається через „X1“ (струмова петля) або „X2“ (RS485) роз'єм (типу RJ25, шість контактів). Для розширення системи зчитування даних (наприклад, при підключенні додаткового MCL 5.8) використовується роз'єм CLin (вхід струмової петлі).

Блок-схема контролера може бути модифікована залежно від величини (ємності) автоматизованої системи збирання даних та типів використовуваних у ній приладів

#### 4.4.3 Основні характеристики інтегрованого GSM/GPRS модуля

Межі робочої температури, оС	від - 30 до + 85
Частота мережі GSM, МГц	Чотирьохчастотна: 850 / 900 / 1800 / 1900 МГц
Клас GPRS	Клас 10 (4RX/2TX, Max 5 Slots) *
Схеми GPRS коду	CS1...CS4
Канал пакетів GPRS	PBCCN
Класи передачі потужності	Діапазон Потужність

Чутливість, dBm хв.	1800/1900 Клас1 (1 Вт)
Роз'єм антени	850/900 Клас4 (2 Вт)
Роз'єм SIM картки,	
Основні засоби управління	-102
Протоколи внутрішнього GPRS модему	SMA
Передача CSD даних	1.8/3.3
Формат GSM SMS	AT команди, система меню
Сертифікати	TCP/IP, UDP

#### 4.4.4 Основні модулі контролера:

а) Модуль модему GSM/GPRS

б) Роз'єми комунікації без гальванічної розв'язки – один або кілька (зазначається при замовленні):

- RS232 для локального програмування MCL5.8 та зчитування даних комп'ютером;

- CL 20 mA „струмова петля“: дозволяє підключити до 3 лічильників електроенергії (залежно від довжини провідників струмової петлі);

- Двопровідний RS485 роз'єм дозволяє паралельно підключити до 32 лічильників електроенергії;

- Додатковий роз'єм для підключення контролера ADNS: CLin "пасивна струмова петля" (гальванічно відв'язана) або RS485 роз'єм;

с) Мікропроцесорне ядро з оперативною пам'яттю ємністю 4 кб та „flash“ пам'яттю ємністю 128 кб для зберігання програм;

д) Імпульсне джерело живлення AC (90-264)/DC 5В 2А;

е) Внутрішній літєво-полімерний акумулятор ємністю 450мАч для забезпечення передачі повідомлення про зникнення напруги або підтримки функції передачі.

Акумулятор заряджається від внутрішнього або зовнішнього джерела живлення контролера.

Варіант 1) При зникненні напруги живлення контролер користується зарядом акумулятора та надсилає повідомлення про відсутність напруги. Після надсилання повідомлення контролер автоматично вимикається і знову вмикається тільки після відновлення напруги живлення. (Детальний опис далі)

Варіант 2) Якщо напруга живлення АС зникає, функція передачі даних підтримується, використовуючи внутрішній акумулятор.

Після закінчення терміну служби акумулятора він може бути замінений без паяння;

f) Антена типу SMA. Зовнішня магнітна антена GSM з кабелем довжиною 2.5...5 м може бути замовлена під час замовлення контролера. Антена може бути 2.5...11 dBi спрямованого посилення.

#### 4.4.5 Технічні характеристики контролера MCL 5.8

Межі напруги живлення (50/60Гц) Споживана потужність: а. Максимальна в режимі передачі: б. Середня у черговому режимі:	90 ... 264 В <2.5 Вт (6ВА)  <1.7 Вт (5ВА)
Напруга у відкритій "струмній петлі", Струм у „струмовий петлі“, мА:	13...15 18 ... 20
Швидкість обміну даними, лічильник-контролер, бод Швидкість обміну даними на вході роз'єму струмової петлі	1200...9600

(пасивна), бод	1200...9600
Швидкість обміну даними, ПК-контролер через RS232 роз'єм, (8N1 байти) бод	4800 / 9600
Кількість лічильників, що підключаються до роз'єму 20 мА „струмової петлі“ (залежить від довжини провідників)	1... 3
Кількість лічильників, що підключаються до RS485 роз'єму	1... 32
Максимальна довжина ліній, що рекомендується, „струмова петля“/RS485 ,м	30
Кількість сесій одночасного підключення через GSM/CSD/GPRS	1
Тривалість чергового режиму від внутрішнього акумулятора менше (при +25 оС) *	5 час.
Тривалість повного заряду внутрішнього акумулятора*	3 час.
Кількість циклів заряду/розряду внутрішнього акумулятора*	300
Робоча температура навколишнього середовища, ОС	-25 ... +55
Температура зберігання, оС	
- модифікації без акумулятора	-40 ... +70
- модифікації з акумулятором	-40 ... +50
Відносна вологість робочого	5 ... 95

середовища, %	
Габарити, мм	93 x 68 x 37

## **4.5 Багатофункціональний лічильник активної, реактивної та повної енергії**

### **EPQS**

#### **4.5.1 Призначення та особливості лічильника**

Електронний багатофункціональний лічильник електроенергії EPQS призначений для обліку активної, реактивної та повної енергії у трифазних мережах змінного струму. Лічильник може вести як однотарифний, і багатотарифний облік. Лічильник вимірює, реєструє та накопичує дані про активну енергію обох напрямків (+A, -A), реактивну енергію обох напрямків (+R, -R), реактивну енергію в чотирьох квадрантах (R1, R2, R3, R4, а також +R і -R) та повної енергії в обох напрямках (+W, -W). Лічильник реєструє максимуми потужності, формує профілі навантаження та реєструє кумулятивні потужності.

EPQS має 16 вільно програмованих каналів для формування профілів вибраних величин, використовуючи вказаний алгоритм формування. Лічильник може виконувати функції аналізу якості енергії та формувати тижневі звіти якості енергії. Для багатотарифного обліку можна використовувати до восьми тарифів для енергії та стільки ж – для максимумів потужності. Структура тарифного модуля дозволяє ввести у лічильник практично будь-яку існуючу тарифну програму. Лічильник оснащений двома незалежними інтерфейсами електричного зв'язку, призначеними для дистанційного зчитування даних. Для локального зчитування даних призначено інтерфейс оптичного зв'язку D0.



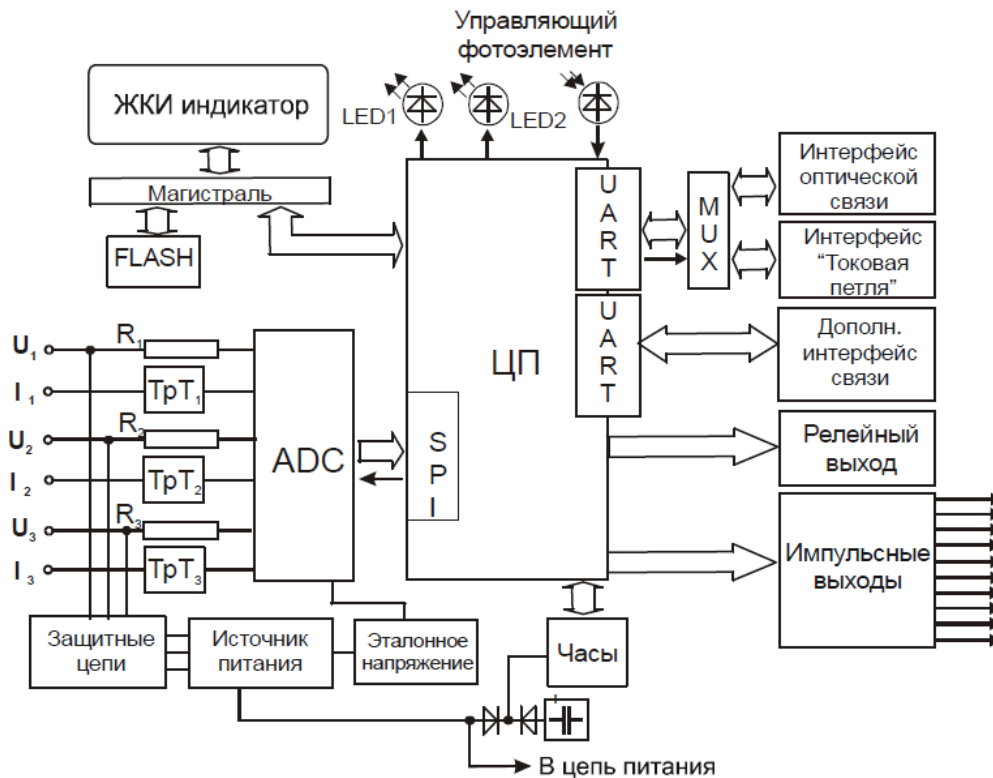


Рис.4.8 Структурна схема лічильника

#### 4.5.2. Функціонал лічильника EPQS

##### 1. Вимірювальний модуль.

Вимірювальний модуль лічильника формує аналогові сигнали, пропорційні величинам напруги та струму. Сигнал напруги формується резистивним дільником напруги, а сигнал струму прецизійним трансформатором струму. Зсув фаз, що з'являється в ланцюзі вимірювання струму, компенсується програмою. Струм і напруга кожної фази вимірюються окремим модулем вимірювальним (далі в тексті «вимірювальний елемент»).

##### 2. Перетворення аналогових сигналів.

Аналогові сигнали струму і напруги вимірювального модуля подаються в шестиканальний ADC (Analog to digital converter) аналого-цифровий перетворювач сигналів (Інтегратор Сигма дельта). У ньому аналогові сигнали перетворюються в шістнадцятибітовий цифровий код 72 рази за один період

мережі. Логічний помножувач сигналів поєднує сигнали струму та напруги різних фаз у єдину послідовність цифрового коду.

### **3. Центральний процесор**

Цифрові коди передаються центральний процесор, де множаться на калібрувальні константи, введені в лічильник під час калібрування. З отриманих значень процесор вираховує квадрати напруги і струму кожної фази, і квадрати напруги, фази яких повернені на 900 (необхідно для розрахунку реактивної енергії):  $IA^2$ ,  $IB^2$ ,  $IC^2$ ,  $UA^2$ ,  $UB^2$ ,  $UC^2$ ,  $UA'$ ,  $UB'$ ,  $UC'$ . З цих величин обчислюються енергії потужності активної, реактивної та повної енергій.

Основний процесор також управляє пристроями пам'яті, РКІ, роботою інтерфейсів зв'язку, формує калібрувальні імпульси (LED) та виконує інші функції.

### **4. Влаштування постійної пам'яті**

Усі дані лічильника, крім миттєвих значень, зберігаються у пристрої постійної пам'яті. Це енергонезалежна пам'ять типу FLASH. Дані в неї записуються наприкінці кожного періоду інтегрування, наприкінці доби та місяця, а також у разі відключення мережі.

### **5. Внутрішній годинник**

Автономний внутрішній годинник лічильника вважає реальний час (годинник, хвилина, секунда), дату (рік, місяць, день, день тижня) і формують керуючі сигнали для перемикавання тарифів. У разі відключення мережі годинник живиться від резервного джерела живлення. При вимкненому напрузі мережі, годинник зберігає функцію управління тарифним модулем і вважає час не менше десяти років. У разі виходу з ладу годинника лічильник працює як одготарифний прилад, а дані накопичуються в реєстрі тієї тарифної зони, яка була вказана під час параметризації лічильника (будь-який з активованих тарифів енергії та потужності). Корекцію годинника можна зробити під час параметрування лічильника. Допускається корекція в межах від -50 до +50 секунд. Годинник лічильника

може автоматично переходити на зимовий/літній час. Дати та час переходу та повернення встановлюються при параметрі лічильника. Час вказується у форматі ММ-DD-НН (місяць – день – година). Також встановлюється крок переведення у годиннику. Можливі наступні варіанти переходу на зимовий/літній час:

- Вказуючи місяць, день, годину та крок перекладу. У вказаний час годинник переводиться на вказаний крок.
- Вказується місяць та день. У цьому випадку годинник із зазначеним кроком переводиться вперед о другій годині ночі при переході на літній час і повертаються о третій годині ночі при переході на зимовий час.
- Вказується лише місяць. У цьому випадку при переході до літнього часу, години о другій годині ночі останньої неділі обраного місяця переводяться вперед із зазначеним кроком. А повертаючись на зимовий час о третій годині ночі в останню неділю обраного місяця переводяться з тим самим кроком назад.
- Час переходу/повернення не вказано. У цьому випадку перехід на літній час відбувається о другій годині ночі останньої неділі березня. І повертається на зимовий час о третій годині ночі останньої неділі жовтня.
- Заборона на зміну літнього/зимового часу. У цьому випадку годинник не переводиться.

#### **4.5.3. Рідкокристалічний індикатор. (РКІ)**

Лічильник має буквено-цифровий чотирирядний 64-символьний рідкокристалічний індикатор для відображення даних. Індикатор можна умовно поділити на 13 інформаційних полів (зон). (див. рис.4.9)

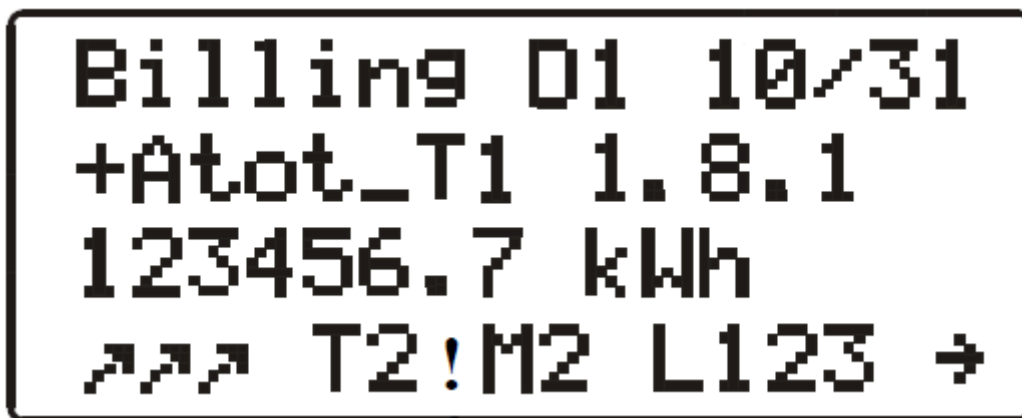


Рис.4.9 Інформаційні поля індикатора лічильника.

В даному розділі був даний короткий опис призначення та принципів побудови систем моніторингу сонячних станцій, складено стислий опис вимог до системи, що розроблялася, і принципів її побудови.

Дати опис структури інформаційних потоків та функціональної структури.

Був даний опис інтерфейсів та протоколів, а також основного обладнання: інвертори струму, інтелектуальні лічильники, контролер, модеми.

## Розділ 5. Розробка програмного забезпечення

Програмне забезпечення розроблено за допомогою об'єктно-орієнтованої мова програмування Java яка є незалежною від архітектури тому, що програма, написана на мові Java, працюватиме на будь-якій підтримуваній апаратній чи системній платформі без змін у початковому кодї та перекомпіляції

Програмне забезпечення системи прогностичного моніторингу поділено на шість окремих програмних блоків (додаток А1.1):

1. Визначення інтенсивності сонячного випромінювання на поверхні модуля  $S(t)$  (додаток А1.2).
2. Отримання квадратичного прогнозу  $Q(t)$  (додаток А1.3)
3. Отримання довірчого прогнозу  $T(t)$  (додаток А1.4)
4. Отримання рефлексивного прогнозу  $R(t)$  (додаток А1.5)
5. Порівняння прогнозів (кумулятивний ансамбль) (додаток А1.6)
6. Динамічна корекція прогнозу (додаток А1.6)

Для визначення інтенсивності сонячного випромінювання на поверхні модуля  $S(t)$  потрібні початкові дані:

- кут нахилу модуля;
- полярний кут, на який повернутий модуль;
- широта розташування модуля;
- довгота розташування модуля;

За допомогою цих даних у програмі розраховуються всі необхідні складові для визначення інтенсивності сонячного випромінювання на поверхні модуля  $S(t)$ : кут між екватором та уявною лінією, що з'єднує центри Землі та Сонця, місцевий стандартний часовий меридіан, рівняння години у хвилинах  $EoT$ , часового поправного коефіцієнта  $TC$ , місцевої сонячної години  $LST$ , часового куту  $HRA$ , кута піднесення  $\alpha$ , атмосферної маси  $AM$  та полярного кута сонця  $\Theta$ .

Інтенсивність сонячного випромінювання є величиною для визначення трьох типів прогнозів.

У квадратичному прогнозі визначаються функції впливу відповідно до усіх погодних даних, що використовуються для прогнозування, які дозволяють отримати систему лінійних рівнянь, рішення якої дозволить отримати прогноз генерації електроенергії.

Методом довірчого прогнозу є визначення умови неправдоподібного прогнозу, за допомогою якої будуть видалені неправдоподібні дані з навчальної послідовності

За допомогою рефлексивного прогнозу прогнозні дані корегуються за станом хмарності

Після отримання усіх прогнозів проводимо порівняння результатів та визначаємо найкращий. Порівняння результатів відбувається за допомогою порівняння прогнозів кумулятивного ансамблю. Та на при кінці програма виконує динамічну корекція прогнозу де порівнюються дійсна попередня генерація та прогноз, та визначається відповідна корекція для поточного прогнозу.

### **Економічне обґрунтування**

Системи відновлюваних джерел енергії набирають дедалі більшої популярності щодня у зв'язку зі своєю автономністю та невичерпністю. Істотним фактором, що впливає на вироблення електричної енергії цих типів джерел, є їх залежність від погодних умов Знання про можливу генерацію електрики дало б можливість точно розрахувати роботу вітропарків чи сонячних електричних станцій при їх проектуванні, що, в свою чергу, дало б перевага в стабільності енергії, що виробляється, доступність інформації про можливість покриття потужності в піковий годинник з метою зменшення навантаження на об'єкти, які використовують традиційні джерело енергії.

Однак для якісного прогнозу генерації енергії необхідні точні та якісні параметри погодних умов.

У цьому дипломному проекті пропонується створення програмного продукту для моніторингу, збору інформації та прогнозування виробки електроенергії сонячними електростанціями.

В економічному розділі пропонується розрахунок трудомісткості обробки програмного продукту, а також кількість всіх витрат, які потрібні при створення програмного коду.

### **6.1 Визначення трудомісткості обробки програмного продукту**

Визначення трудомісткості обробки програмного продукту. Нормування праці у процесі створення програмного забезпечення суттєво ускладнено через творчий характер роботи програміста. Тому, трудомісткість обробки програмного забезпечення може бути розраховано з урахуванням систем моделей з різною точністю оцінки.

Трудомісткість обробки програмного забезпечення можна розрахувати

за формулою:  $t = t_0 + t_i + t_a + t_p + t_{\text{опл}} + t_d = 8 + 2.16 + 7.02 + 7.02 + 36.85 + 16.11 = 77.16$  людино - годин,

де  $t_0$  – витрати на підготовку та опис поставленого завдання;

$t_i$  – витрати на дослідження алгоритму рішення завдання;

$t_a$  – витрати на обробку блок-схеми алгоритму;

$t_p$  – витрати на програмування за готовою блок схемою;

$t_{\text{опл}}$  – витрати на налаштування програми на електроннообчислювальній машині;

$t_d$  – витрати на підготовку документації за завданням.

Оцінка витрат праці на підготовку та опис завдання залежить від конкретних умов та визначається на основі експертних оцінок. Для даного дипломного проекту  $t_0 = 8$  людино – годин. Витрати праці вивчення опису завдання визначаються з урахуванням уточнення опису та кваліфікації програміста за формулою:

$$t_i = Q \cdot B (75 \dots 85) \cdot k = 117.936 \cdot 1.175 \cdot 0.8 = 2.16 \text{ людино – годин,}$$

де  $B$  – коефіцієнт збільшення витрат праці

$k$  – коефіцієнт кваліфікації програміста, який визначається

залежно від стажу роботи за фахом. Для цього випадку  $k = 0.8$  (в

даному випадку досвід до 2 років);

$p$  – коефіцієнт корекції програми у процесі її обробки

Витрати праці на обробку алгоритму вирішення завдання:

$$t_a = Q(20 \dots 25) \cdot k = 117.93621 \cdot 0.8 = 7.02 \text{ людино – годин.}$$

Витрати праці на складання програми за готовою блок-схемою:

$$t_{п} = Q(20 \dots 25) \cdot k = 117.93621 \cdot 0.8 = 7.02 \text{ людино – годин.}$$

Витрати праці на налаштування програми на електронно-обчислювальній машині розраховується за формулою:

$$t_{опл} = Q4 \cdot k = 117.9364 \cdot 0.8 = 36.85 \text{ людина – годин.}$$

Витрати праці в підготовку документації за завданням визначаються за формулою:

$$t_{Д} = t_{ДР} + t_{ДО} = 9.21 + 6.9 = 16.11 \text{ людино – годин,}$$

де  $t_{ДР}$  – трудомісткість підготовки матеріалів до рукопису:

$$t_{ДР} = Q16 \cdot k = 117.93616 \cdot 0.8 = 9.21 \text{ людино – годин;}$$



$t_{ДО}$  – трудомісткість редагування, друку та оформлення документації:

$$t_{ДО} = 0.75 \cdot t_{ДР} = 0.75 \cdot 9.21 = 6.9 \text{ людино – годин,}$$

## 6.2 Розрахунок витрат за створення програмного продукту

Витрати створення програмного продукту Кпо включають витрати на заробітну плату виконавця програми Ззп та машинного часу, необхідного для налаштування програми на електронно-обчислювальній машині Змв:

$$Кпо = Ззп + Змв = 9645 + 128 = 9773 \text{ грн.}$$

Заробітна плата виконавця визначається за такою формулою:

$$Ззп = t \cdot Спр = 77.16 \cdot 125 = 9645 \text{ грн.,}$$

де  $t$  - загальна трудомісткість обробки програмного забезпечення;

$Спр = 125 \text{ грн./год}$  - середня годинна заробітна плата програміста з нарахуваннями (для міста Дніпро з урахуванням малого досвіду молодшого розробника).

Вартість машинного часу, необхідного для налаштування програми на електронно-обчислювальній машині:

$$Змв = t_{опл} \cdot С_{мч} = 60.48 \cdot 2.12 = 128 \text{ грн.,}$$

где  $t_{опл}$  – трудомісткість відладки програми на електронно-обчислювальній машині;

$С_{мч}$  – ціна машино-годин електронно-обчислювальній машині, грн./год.

Ціна 1 години машиного часу ПК, котора знаходиться за формулою:

$$C_{мч} = P \cdot t_{ПК} \cdot C_e + \Phi_{зал} \cdot Ha \cdot Fr_{г} + K_{лпз} \cdot Ha_{пз} \cdot Fr_{г} = 0,05 \cdot 1 \cdot 1.92072 \cdot 1.2 + 9999 \cdot 0,4 \cdot 1993 + 0 \cdot 0,4 \cdot 1993 = 2.12 \text{ грн.}$$

де  $P = 0,05$  – установлена міць ПК, кВт;

$C_e = 192.072$  – тариф на електроенергію, коп/кВт\*год (для споживачів 2 категорії, станом на червень 2018 року);

$t_{ПК}$  – час роботи ПК

$\Phi_{зал} = 9999$  грн – залишкова вартість ПК при купівлі в серпні місяці 2017 року (Ноутбук марки «DELL», моделі «Latitude 3540»)

$Ha = 0,4$  – річна норма амортизації на ПК, частини одиниці (станом на червень 2018 року);

$Ha_{пз} = 0,4$  – річна норма амортизації на ліцензійне ПЗ, частини одиниці на час 2018 року ;

$K_{лпз} = 0$  грн, вартість ліцензійного програмного забезпечення

$Fr_{г}$  – річний фонд робочого часу (при 40-ти годинному робочому тижні  $Fr_{г} = 1993$  годин). Визначені в такий спосіб витрати на створення програмного забезпечення є частиною одноразових капітальних витрат на створення автоматизованої системи управління підприємством або іншої апаратури автоматизації. Очікувана тривалість обробки програмного забезпечення:

$$T = t_{BK} \cdot Fr_{п} = 77.161 \cdot 176 = 0.438 \text{ мес.}$$

где  $BK$  – число виконавців;

$Fr_{п}$  – місячний фонд робочого часу (при 40 годинному робочому тижні  $Fr_{п} = 176$  годин).

### 6.3 Висновки у розділі

У цьому розділі дипломного проекту було розраховано витрати під час розробки та використання розглянутого програмного продукту. У сумі загальні витрати:

$$K_{\text{по}} = Z_{\text{п}} + Z_{\text{мв}} = 9645 + 128 = 9773 \text{ грн.}$$

Витрати в такій сумі зазнає потенційний користувач, що займається програмуванням метеостанції даного типу на основі програмного коду.

## **ОХОРОНА ПРАЦІ ТА ЗАХИСТ НАСЕЛЕННЯ В НАДЗВИЧАЙНИХ СИТУАЦІЯХ**

### **7.1 Основи охорони праці**

Охорона праці – це наукова дисципліна, яка вивчає теоретичні і практичні питання безпеки праці, причини виробничого травматизму і розробляє заходи щодо створення здорових і безпечних умов праці. Головним об'єктом її досліджень є людина в процесі праці, виробниче середовище і обставини, взаємозв'язок людини з промисловим обладнанням, з технологічним процесом. Для вирішення цих питань використовуються наукові досягнення багатьох галузей знань: соціально-правових і економічних наук – трудове право і економіка; медичних наук – гігієна праці, психологія праці, фізіологія праці, промислова токсикологія та ін; технічних наук – загальноінженерних і профільюючих дисциплін, інженерна психологія, промислова етика.

В зв'язку з тим, що деякі галузі народного господарства мають специфічні умови праці, виникає потреба розробляти галузеві правила і норми охорони праці .

Нормативно-технічна документація з охорони праці є основою для розробки заходів щодо забезпечення на всіх місцях безпечних і здорових умов праці. На кожному підприємстві з урахуванням вимог згаданих нормативотехнічних документів розробляють інструкції з охорони праці з урахуванням конкретних умов для кожної робочої професії.

Серед нормативно-технічної документації, яка ставить вимоги з охорони праці, важливе місце посідає стандартизація. Система стандартів безпеки праці (ССБП) – це комплекс взаємопов'язаних стандартів, спрямованих на забезпечення безпеки праці, збереження здоров'я і працездатності людини в процесі праці. Робота із стандартизації вимог до

безпеки праці ведеться у двох напрямках: розробка спеціальних стандартів (ССБП) та включення до стандартів і технічних умов спеціального розділу "Вимоги безпеки".

Контроль за виконанням законодавства про працю, правил і норм техніки безпеки, виробничої, санітарної і протипожежної безпеки ведеться по трьох напрямках: державному, внутрівідомчому і громадському.

Державний нагляд і контроль здійснюють спеціалізовані органи нагляду, які не залежать в своїй діяльності від адміністрації підконтрольних організацій і їх вищестоячих органів; технічні і правові інспекції, що підпорядковані профспілковим органам.

Державний нагляд за станом охорони праці здійснюють Державний комітет України з охорони праці, Державний комітет України з ядерної та радіаційної безпеки, органи Державного пожежного нагляду, органи санітарно-епідеміологічної служби.

Державний санітарний нагляд здійснюється санітарноепідеміологічними станціями, які контролюють санітарний стан промислових об'єктів, громадських і житлових будівель, розробляють санітарно-гігієнічні нормативи, які є обов'язковими при проектуванні будівель і експлуатації підприємств.

Санітарні інспекції проводять облік і розслідування причин професійних захворювань, беруть участь в роботі по зниженню загальних і професійних захворювань.

Державний пожежний нагляд в системі Міністерства внутрішніх справ веде велику профілактичну роботу: здійснює нагляд за дотриманням протипожежного режиму в процесі проектування будівництва і експлуатації підприємств, установ і організацій; здійснює прийом побудованих і реконструйованих об'єктів, обстежує в процесі експлуатації всі об'єкти з метою перевірки виконання протипожежних вимог, розробляє протипожежні

правила і норми, контролює їх виконання, дає консультації з питань встановлення систем пожежної безпеки підприємств, вирішує питання про організацію пожежної охорони різноманітних об'єктів народного бб господарства, стежить за їх устаткуванням, бере участь в розслідуванні пожеж і бере їх на облік, вживає заходи щодо усунення причин пожежі.

Державний і громадський контроль і нагляд в галузі охорони праці не знімає відповідальності з керівників і власників підприємства за охорону праці.

Законом України «Про охорону праці» встановлено відповідальність посадових осіб за порушення правил охорони праці. Відповідальність може бути дисциплінарна, адміністративна, кримінальна і матеріальна [12].

Дисциплінарну – накладають у випадках, коли з вини посадових осіб, власника, адміністративно-технічних і адміністративно-господарських працівників допускаються порушення охорони праці, які не призводять до тяжких наслідків. На посадову особу в порядку підлеглості накладають такі стягнення: догана, звільнення з роботи.

Адміністративна відповідальність постає тоді, коли особи адміністративно-управлінського персоналу порушили правила охорони праці в будівництві, санітарно-гігієнічні правила, правила пожежної безпеки, правила охорони атмосферного повітря, землі, надр, лісів, водних ресурсів. Підприємство чи організація сплачує штраф за кожний нещасний випадок та випадок професійного захворювання, які сталися на виробництві з його вини.

Кримінальна відповідальність полягає в покаранні осіб, які допустили порушення правил охорони праці, що могли призвести або призвели до нещасних випадків чи інших тяжких наслідків.

Матеріальна відповідальність полягає у тому, що органи соціального страхування і соціального забезпечення мають право стягувати з підприємства суму пенсії, яка була виплачена потерпілому при нещасному

випадку з вини адміністрації. Доказом вини є акт про нещасний випадок, пов'язаний з виробництвом, постанова слідчих органів і звинувачувальний вирок суду.

Організація і координація робіт в галузі охорони праці повинна передбачати формування органів управління охороною праці, встановлення 67 обов'язків і порядку взаємодії осіб, які беруть участь в управлінні, а також прийнятті і реалізації управлінських рішень.

Планування робіт з охорони праці повинно включати визначення завдань підрозділів і служб підприємства. Контроль за станом охорони праці повинен бути спрямований на перевірку стану умов праці, виявлення відхилень від вимог стандартів ССБП, норм і правил з охорони праці, перевірку виконання служба і підрозділами своїх обов'язків у галузі охорони праці, на прийняття ефективних заходів щодо ліквідації недоліків. Інструктаж працюючих поділяють на вступний, початковий, на робочому місці, повторний, позаплановий і поточний.

Вступний інструктаж з усіма, хто поступає на роботу незалежно від їх освіти і стану роботи по даній професії, проводить інженер з охорони праці за затвердженою програмою, головним інженером підприємства та з обов'язковим підписом тих, хто проводив і отримував інструктаж.

Початковий інструктаж на робочому місці, повторний, позаплановий і поточний проводить керівник робіт. Початковий інструктаж на робочому місці проводять при прийомі на роботу нових робітників за інструкцією з охорони праці, розробленою для окремих професій або видів робіт.

Повторний інструктаж проходять всі працівники незалежно від кваліфікації, освіти і стажу роботи не рідше ніж через шість місяців. Його проводять з метою перевірки знання робітниками правил і норм з охорони праці.

Позаплановий – проводять, коли змінилися правила охорони праці або технологічний процес, обладнання, інструмент, сировина та інші фактори, що впливають на безпеку праці.

Поточний інструктаж проводять перед виконанням робіт, на які необхідно оформити наряд-допуск. Проведення його фіксується в наряді допуску на виконання робіт. 68

## **7.2 Захист населення у надзвичайних ситуаціях**

Техногенне, екологічне та природне становище України рік від року стає складнішим, зростає його негативний вплив на населення та навколишнє середовище. Тому, з набуттям Україною незалежності, враховуючи досвід економічно розвинутих країн, було розпочате законодавче оформлення Цивільної оборони, як державної системи органів управління та сил для організації і здійснення заходів щодо захисту населення від впливу надзвичайних ситуацій.

В 1993 році було прийнято закон України “Про цивільну оборону”. Держава, як гарант цього права, створює систему Цивільної оборони, яка має на меті захист населення від небезпечних наслідків аварій, катастроф техногенного, екологічного, воєнного характеру.

Захист населення від наслідків стихійних лих, аварій, катастроф, вибухів, пожеж і застосування засобів ураження здійснюється проведенням комплексу заходів, які повинні забезпечити укриття населення в захисних спорудах, його евакуацію, медичний, радіаційний і хімічний захист від впливу біологічних засобів ураження.

Потреба у захисних спорудах визначається, виходячи з необхідності укриття всіх працюючих за місцем роботи і проживання всього непрацюючого населення за місцем проживання.

На території територіальних громад часом знаходяться небезпечні об’єкти. Це склади, на яких зберігають хімічні речовини, які використовують в сільськогосподарському виробництві. Дані об’єкти слід обробити захисним



матеріалом. В разі якогось стихійного лиха можна сховатись в сховища, які є на території Ременівської сільської ради і вони можуть служити укриттям.

Крім складів є лінії електропередач, біля яких можна дати санітарнозахисну зону, де є обмежене використання земель. Навколо газопровідних об'єктів слід ставити огорожу і розпізнавальний знак, що попереджає про небезпеку.

На території торфорозробок можуть виникати пожежі. Тому населення, яке живе поблизу, повинне про це пам'ятати і якщо це може загрожувати життю, то слід евакуувати з небезпечної зони в безпечну.

Керівники підприємств є відповідальними за захист населення, а особливо працівників, які працюють в даних господарствах. Тому вони повинні дотримуватись правил техніки безпеки. Керівники підприємств повинні розробити плани по евакуації населення.

Слід увагу приділити керівникам щодо індивідуального захисту працівників, які працюють на небезпечних об'єктах. Це зокрема можуть бути комбінезони, рукавиці і щонайпростіше марлеві пов'язки, які забезпечать захист слизових оболонок і шкіри.

З метою покращення стану охорони праці, недопущень виробничого травматизму, професійних захворювань необхідно постійно здійснити наступні заходи:

- регулярно поводити інструктажі по техніці безпеки і вести їх чіткий облік;
- суворо дотримуватись вимог і правил техніки безпеки при проведенні обмірів земельних ділянок та інших геодезичних робіт;
- забезпечити працюючий персонал необхідним спецодягом, спецвзуттям та іншими засобами;
- запобігати виникненню надзвичайних ситуацій і запровадити заходи, щодо зменшення збитків та втрат у разі аварій чи катастроф;
- створити системи управління (оповіщення, зв'язок, спостереження, контроль);
- захистити населення від наслідків аварій, катастроф, стихійних лих тощо;

- організувати та проводити рятувальні та інші невідкладні роботи в районах лиха.

Дотримання цих вимог дозволить покращити умови охорони праці та захист населення при виконанні господарської і виробничої діяльності.

### **ЗАГАЛЬНІ ВИСНОВКИ**

У дипломному проекті було розглянуто основні способи прогнозування генерації електричної енергії за допомогою сонячних електростанцій, а також їх актуальність.

Було складено методика та алгоритм розрахунку. За розробленою методикою було створено програмне забезпечення для автоматизації процесів курування сонячних станцій. Даний програмний продукт дозволить якісно та точно прогнозувати виробку електроенергії сонячними електростанціями.

В економічній частині дипломного проекту було розраховано витрати пов'язані з трудомісткістю обробки та витрати на створення даного програмного продукту

### **ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ**

- 1) В.В.Вьюгин МАТЕМАТИЧЕСКИЕ ОСНОВЫ ТЕОРИИ МАШИННОГО ОБУЧЕНИЯ И ПРОГНОЗИРОВАНИЯ /2013
- 2) Арженовский С.В., Молчанов И.Н. Статистические методы прогнозирования/2001
- 3) Тюньков Д.А., Сапилова А.А., Грицай А.С., Алексеенко Д.А., Хамитов Р.Н., Методы краткосрочного прогнозирования выработки электрической энергии солнечными электростанциями и их классификация
- 4) Day-Ahead Hourly Forecasting of Power Generation from Photovoltaic Plants Lorenzo Gigoni, Alessandro Betti, Emanuele Crisostomi/2019
- 5) Басок Б.И., чл.-кор. НАН Украины, Давыденко Б.В., докт. техн. наук, Новиков В.Г., канд. техн. наук, Сороковой Р.Я. МОДЕЛИРОВАНИЕ СОЛНЕЧНОГО ИЗЛУЧЕНИЯ ДЛЯ ИСПОЛЬЗОВАНИЯ В

ВОЗОБНОВЛЯЕМОЙ ЕНЕРГЕТИКЕ И В СТРОИТЕЛЬНОЙ  
ТЕПЛОФИЗИКЕ

- 6) Опис економічного стану альтернативних джерел енергії в 2022р/2022  
[https://api.solarpowereurope.org/uploads/Solar\\_Power\\_Europe\\_Global\\_Market\\_Outlook\\_report\\_2022\\_2022\\_V2\\_07aa98200a.pdf](https://api.solarpowereurope.org/uploads/Solar_Power_Europe_Global_Market_Outlook_report_2022_2022_V2_07aa98200a.pdf)
- 7) Положення про навчально-методичне забезпечення освітнього процесу Національного технічного університету «Дніпровська політехніка» / Укладачі: Ю.О. Заболотна, Є.А. Коровяка, В.О. Салов; М-во освіти і науки України, Нац. техн. ун-т. «Дніпровська політехніка» – Д. : НТУ «ДП», 2018. – 23 с.
- 8) Положення про організацію атестації здобувачів вищої освіти НТУ «Дніпровська політехніка» / Укладачі: Ю.О. Заболотна, О.О. Конопльова, В.О. Салова, В.О. Салов; М-во освіти і науки України, Нац. техн. ун-т. «Дніпровська політехніка» – Д. : НТУ «ДП», 2018. – 40 с.
- 9) Стандарт вищої освіти України. Рівень вищої освіти перший (бакалаврський) рівень. Ступінь вищої освіти бакалавр. Спеціальність 151 Автоматизації та комп'ютерно-інтегровані технології. МОН України. – Київ. – 2018. – 17 с.
- 10) ДСТУ 3008:2015. Звіти у сфері науки і техніки. Структура та правила оформлювання / Нац. стандарт України. – Вид. офіц. – [чинний від 2017-07-01]. – Київ : ДП «УкрНДНЦ», 2016. – 27 с.
- 11) ДСТУ 1.5:2015. Правила розроблення. Викладання та оформлення національних нормативних документів оформлювання / Нац. стандарт України. – Вид. офіц. – [чинний від 2017-02-01]. – Київ: ДП «УкрНДНЦ», 2016. – 61 с.
- 12) 1 ДСТУ 8302:2015. Бібліографічне посилання. Загальні положення та правила складання / Нац. стандарт України. – Вид. офіц. – [Уведено вперше ; чинний від 2016-07-01]. – Київ : ДП «УкрНДНЦ», 2016. – 17 с.

- 13) 1 Бойко О.О. Методичні вказівки до лабораторних робіт з теорії автома-тичного управління для студентів напрямку підготовки «Комп'ютерна інжене-рія» / Укл.: О.О. Бойко – Д.: Державний ВНЗ «НГУ», 2017. – 107 с. – Режим доступу: <https://goo.gl/nUMtFE>. – Назва з домашньої сторінки Інтернету.
- 14) ДСТУ 5003.1 Автоматизовані системи обліку електричної енергії. Загальні положення
- 15) ДСТУ 5003.2 Автоматизовані системи обліку електричної енергії. Терміни та визначення понять
- 16) ДСТУ 5003.3-1 Автоматизовані системи обліку електричної енергії. Структура, функції та види забезпечення
- 17) ДСТУ 5003.4-1 Автоматизовані системи обліку електричної енергії. Забезпечення сумісності. Основні положення

## Додаток А

### Програмне забезпечення системи керування

#### А.1 Програмне забезпечення об'єкта

##### А.1.1 Список програм

Основна програма (Main)

Квадратичний прогноз Q(t)

Довірчий прогноз T(t)

Рефлексивний прогноз R(t)

Порівняння прогнозів

Динамічна корекція прогнозу

##### А.1.2 Основна програма

```
public void startForecastTask(LocalDateTime ldt) {
    try {
        if (!new InitApp().initMainApp()) {
            throw new Exception("Main application initialization error!");
        } else {
            if (mainConfigParams.configs.isEmpty()) {
                throw new Exception("Config list is empty!");
            } else {
                int skipPointsCount = 0;
                countXMLreport = 0;
                for (String config : mainConfigParams.configs) {
                    try {
                        ConfigParams configParams = new InitApp().initApp(mainConfigParams.pathPrefix + config);
                        if (configParams == null) {
                            throw new Exception("Program initialization error!");
                        } else {
                            LOG.info("Job for " + config + " starts");
                            if (!isStartStudy && configParams.schedule != null) {
                                String hour = ldt.format(DateTimeFormatter.ofPattern("HH:mm"));
                                //hour = "07:00";
                                if (isTomorrow && !isToday) {
                                    if (!configParams.schedule.correction.tomorrow.contains(hour)) {
                                        configParams.xmlParameters.isScipReport = true;
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```

    } else if (isToday && !isTomorrow) {
        if (!configParams.schedule.correction.today.contains(hour)) {
            configParams.xmlParameters.isScipReport = true;
        }
    }
}

// Gson gson = new Gson();
// //Gson gson = new GsonBuilder().setPrettyPrinting().create();
// try (FileWriter writer = new FileWriter("template.json")) {
//     gson.toJson(configParams, writer);
// } catch (IOException e) {
//     e.printStackTrace();
// }

if (configParams.xmlParameters.isScipReport) {
    skipPointsCount--;
}

WorkWithDB db = new WorkWithDB(configParams);
ZoneId zoneId = ZoneId.of(configParams.parameters.timeZone);
if (configParams.parameters.serve != null && !configParams.parameters.serve.generation) {
    TreeMap<LocalDateTime, Double> updatePowerList = new TreeMap<>();
    for (int i = 1; i < 25; i++) {
        updatePowerList.put(ldt.toLocalDate().atStartOfDay().plusHours(i), 0.0);
    }
    if (configParams.parameters.serve.consumption) {
        TreeMap<LocalDateTime, Double> powerList = new TreeMap<>();
        db.getLastPowerData(powerList,
            ldt.toLocalDate().atStartOfDay().atZone(zoneId).toEpochSecond(),
            configParams.parameters.serve.consumption);
        if (powerList.size() != 0) {
            if (powerList.get(powerList.firstKey()) != 0 && powerList.get(powerList.lastKey()) == 0 ||
                powerList.get(powerList.firstKey()) == 0 && powerList.get(powerList.lastKey()) != 0) {
                LocalDate ld = powerList.firstKey().toLocalDate();
                powerList.clear();
                db.getLastPowerData(powerList,
                    ld.minusDays(1).atStartOfDay().atZone(zoneId).toEpochSecond(),
                    configParams.parameters.consumption);
            }
            Duration duration = Duration.between(powerList.firstKey(), updatePowerList.firstKey());
            powerList.forEach((k, v) -> {

```

```

        if (v <= 0) {
            updatePowerList.put(k.plusDays(duration.toDays()), v);
        }
    }
    );
}
}
createXMLReport(configParams, updatePowerList);
} else {
//    temp
//getEmails(configParams, db);
// end temp
    TreeMap<LocalDateTime, ObjectWeather> forecastWeatherData = new TreeMap<>();

    if (cmd.hasOption("forecast")) {
        if (cmd.getOptionValues("forecast").length == 1) {
            LocalDate from = LocalDate.parse(cmd.getOptionValues("forecast")[0],
DateTimeFormatter.ofPattern("dd.MM.yyyy"));
//            db.getForecastData(forecastWeatherData,
//                from.atStartOfDay().format(DateTimeFormatter.ofPattern("dd.MM.yyyy HH:mm")),
//                from.atStartOfDay().plusDays(1).format(DateTimeFormatter.ofPattern("dd.MM.yyyy
HH:mm"))));
            db.getForecastData("WEATHER_HISTORY", forecastWeatherData,
                from.atStartOfDay().plusHours(1).atZone(zoneId).toEpochSecond(),
                from.atStartOfDay().plusDays(1).atZone(zoneId).toEpochSecond());
        } else if (cmd.getOptionValues("forecast").length == 2) {
            LocalDate from = LocalDate.parse(cmd.getOptionValues("forecast")[0],
DateTimeFormatter.ofPattern("dd.MM.yyyy"));
            LocalDate to = LocalDate.parse(cmd.getOptionValues("forecast")[1],
DateTimeFormatter.ofPattern("dd.MM.yyyy")).plusDays(1);
//            db.getForecastData(forecastWeatherData,
//                from.atStartOfDay().format(DateTimeFormatter.ofPattern("dd.MM.yyyy HH:mm")),
//                to.atStartOfDay().format(DateTimeFormatter.ofPattern("dd.MM.yyyy HH:mm"))));
            db.getForecastData("WEATHER_HISTORY", forecastWeatherData,
                from.atStartOfDay().plusHours(1).atZone(zoneId).toEpochSecond(),
                to.atStartOfDay().atZone(zoneId).toEpochSecond());
        }
    }
} else {
//    db.getForecastData(forecastWeatherData,

```

```

//          ldt.toLocalDate().atStartOfDay().format(DateTimeFormatter.ofPattern("dd.MM.yyyy
HH:mm")),
//
ldt.toLocalDate().atStartOfDay().plusDays(1).format(DateTimeFormatter.ofPattern("dd.MM.yyyy HH:mm"));
        db.getForecastData("WEATHER_HISTORY", forecastWeatherData,
                ldt.toLocalDate().atStartOfDay().plusHours(1).atZone(zoneId).toEpochSecond(),
                ldt.toLocalDate().atStartOfDay().plusDays(1).atZone(zoneId).toEpochSecond());
//          db.getForecastData(forecastWeatherData,
//
LocalDate.now().atStartOfDay().minusDays(3).format(DateTimeFormatter.ofPattern("dd.MM.yyyy HH:mm")),
//
LocalDate.now().atStartOfDay().plusDays(3).format(DateTimeFormatter.ofPattern("dd.MM.yyyy HH:mm"));
        }
        if (forecastWeatherData.isEmpty()) {
            throw new Exception("Forecast data is empty!");
        } else {
            createForecast(configParams, forecastWeatherData, 0);
            for (String apiServersTable : mainConfigParams.weather.ApiServersTables) {
                TreeMap<LocalDateTime, ObjectWeather> weatherData = new TreeMap<>();
                db.getForecastData(apiServersTable, weatherData,
                        ldt.toLocalDate().atStartOfDay().plusHours(1).atZone(zoneId).toEpochSecond(),
                        ldt.toLocalDate().atStartOfDay().plusDays(1).atZone(zoneId).toEpochSecond());
                if (weatherData.size() > 0) {
                    createForecastForDiffApiServers(configParams, weatherData, 0, apiServersTable);
                }
            }
        }
        LOG.info("Job for " + config + " finished");
        LOG.info("-----");
    }
} catch (Exception ex) {
    errorHandler(ex, ex.toString());
}
// temp
telegramBot.sendMessage(mainConfigParams.telegram.forecastChannel,
        "всего прогнозируемых точек: " + mainConfigParams.configs.size());
//          telegramBot.sendMessage(mainConfigParams.telegram.forecastChannel,

```



```

//      "исключенно: " + skipPointsCount);
mailList.forEach((k, v) -> new XMLReportSender().sendXMLReports(k, v));
excelReportMailList.forEach((k, v) -> new XMLReportSender().sendExcelReports(k, v));
// end temp
unsignedFilesList.forEach((k, v) -> {
    File file = new File(k);
    File signFile = null;
    int count = 3;
    while (count > 0) {
        if (new JWTSigner().createSing(file.getAbsolutePath(), v)) {
            signFile = new File(file.getAbsolutePath() + ".p7s");
            telegramBot.sendDocument(mainConfigParams.telegram.forecastChannel,
"подписанный файл");
            signFile,
            break;
        }
        count--;
    }
    if (signFile != null) {
        count = 3;
        String reportDate = getReportDate(file);
        while (count > 0) {
            if (new GpeeComUa().putData(v.gpeeComUaAuth.login, v.gpeeComUaAuth.password,
                String.valueOf(v.xmlParameters.code), signFile.getAbsolutePath(),
                reportDate, isUpdateData)) {
                break;
            }
            count--;
        }
    }
    if (signFile != null) {
        String parentDir = signFile.getParent().substring(signFile.getParent().lastIndexOf("\\") + 1);
        String date = LocalDate.parse(parentDir, DateTimeFormatter.ofPattern("yyyyMMdd"));
        format(DateTimeFormatter.ofPattern("dd.MM.yyyy"));
        String mailPath = String.format("\\mail\\%s\\", scheduleHour);
        String mailFileName = Uutils.getMailFileName(signFile.getParent(), mailPath,
v.xmlParameters.wCode, date);
        // powerList.firstKey().toLocalDate(date).format(DateTimeFormatter.ofPattern("dd.MM.yyyy"));
        Uutils.copySignFiles(signFile.getAbsolutePath(), mailFileName);
    }
}

```

```

        });
    }
}
} catch (Exception ex) {
    errorHandler(ex, ex.toString());
}
}
}

```

### A.1.3 Квадратичний прогноз Q(t)

```

public class QuadraticForecast {

    private ConfigParams configParams;

    public QuadraticForecast(ConfigParams configParams) {
        this.configParams = configParams;
    }

    public boolean getQuadraticForecast(ObjectWeatherAndParams owas, int pow, boolean isForecastIrradiation) {
        try {
            // setS_t(owas.getObjectWeather(), isForecastIrradiation);
            // if (cmd.hasOption("cloud")) {
            //     setForecastAveragesValues(owas.getObjectWeather(), isForecastIrradiation);
            // } else {
            //     setAveragesValues(owas.getObjectWeather(), isForecastIrradiation);
            // }
            // removeZeroS(owas.getObjectWeather());
            // removeUnrealPower(owas.getObjectWeather());
            setPow(owas.getObjectWeather(), pow);
            final double[] y = new double[countVars * pow + 1];
            final double[][] a = new double[countVars * pow + 1][countVars * pow + 1];
            setY(y, pow, owas.getObjectWeather());

            // System.out.println(Arrays.stream(y).toArray());
            // for (int i = 0; i < y.length; i++) {
            //     System.out.println("y[" + i + "]" + y[i]);
            // }
            setA(a, pow, owas.getObjectWeather());
            RealVector solutions = getSolution(a, y);
            if (solutions != null) {
                // System.out.println(solutions.toString());
                int countSolutions = 0;
            }
        }
    }
}

```

```

for (int i = 0; i < solutions.getDimension(); i++) {
    ArrayList<Double> solution = new ArrayList<>();
    if (i == 0) {
        solution.add(solutions.getEntry(i));
    } else {
        for (int j = 0; j < pow; j++) {
            solution.add(solutions.getEntry(i + j));
        }
        i = i + pow - 1;
    }
    owas.getSolutions().put(countSolutions, solution);
//    System.out.println(owas.getSolutions().get(countSolutions));
    countSolutions++;
}
setInfluenceFunctions(owas.getSolutions(), owas.getObjectWeather()); //pow,
setForecast(owas.getObjectWeather());
owas.setErrorsAndSumPower();
return true;
}
} catch (Exception ex) {
    errorHandler(ex, ex.toString());
}
return false;
}
}

```

#### **A.1.4 Довірчий прогноз $T(t)$**

```

public class ConfidenceForecast {

    private ConfigParams configParams;

    public ConfidenceForecast(ConfigParams configParams) {
        this.configParams = configParams;
    }

    public boolean getConfidenceForecast(ObjectWeatherAndParams owap, int pow, double cf, double sr) {
        try {
            List<Double> errors = new ArrayList<>();
            TreeMap<LocalDateTime, ObjectWeather> ow = new TreeMap<>();
            owap.getObjectWeather().forEach((k, v) -> {

```

```

        ow.put(k, deepCloneObjectWeather(v));
    });
    TreeMap<LocalDateTime, ObjectWeather> unrealData = getUnrealData(ow, cf);
    //    System.out.println("unrealData.size() = " + unrealData.size());
    double firstCondition = Math.floor(ow.size() * sr);
    //    int countUnrealData = unrealData.size();
    if (unrealData.size() > firstCondition) {
        for (Map.Entry<LocalDateTime, ObjectWeather> entry : unrealData.entrySet()) {
            errors.add(entry.getValue().getModAbsError());
        }
        Collections.sort(errors);
        errors = errors.subList((int) Math.round(errors.size() * (1 - sr)), errors.size());
        for (Map.Entry<LocalDateTime, ObjectWeather> entry : unrealData.entrySet()) {
            if (errors.contains(entry.getValue().getModAbsError())) {
                ow.remove(entry.getKey());
            }
        }
        QuadraticForecast qf = new QuadraticForecast(configParams);
        final double[] y = new double[countVars * pow + 1];
        final double[][] a = new double[countVars * pow + 1][countVars * pow + 1];
        qf.setY(y, pow, ow);
        qf.setA(a, pow, ow);
        RealVector solutions = qf.getSolution(a, y);
        if (solutions != null) {
            setSolutions(solutions, owap, pow);
            qf.setInfluenceFunctions(owap.getSolutions(), owap.getObjectWeather()); //pow,
            qf.setForecast(owap.getObjectWeather());
            return true;
        }
    } else {
        unrealData.forEach((k, v) -> {
            ow.remove(k);
        });
        final double[] y = new double[countVars * pow + 1];
        final double[][] a = new double[countVars * pow + 1][countVars * pow + 1];
        QuadraticForecast qf = new QuadraticForecast(configParams);
        qf.setY(y, pow, ow);
        qf.setA(a, pow, ow);
        RealVector solutions = qf.getSolution(a, y);
    }
}

```

```

        if (solutions != null) {
            setSolutions(solutions, owap, pow);
            qf.setInfluenceFunctions(owap.getSolutions(), owap.getObjectWeather()); //pow,
            qf.setForecast(owap.getObjectWeather());
        }
        //        owap.setErrorsAndSumPower();
        return true;
    }
}
//        owap.setErrorsAndSumPower();
} catch (Exception ex) {
    errorHandler(ex, ex.toString());
}
return false;
}
}
}

```

### A.1.5 Рефлексивный прогноз R(t)

```

public class ReflectiveForecast {

    private final int IDX_SIGMA_SOLUTION = 2;
    private ConfigParams configParams;

    public ReflectiveForecast(ConfigParams configParams) {
        this.configParams = configParams;
    }

    public boolean getReflectiveForecast(ObjectWeatherAndParams owap, int pow, double rf) {
        double sigma1, sigma2;
        try {
            TreeMap<LocalDateTime, ObjectWeather> ow = new TreeMap<>();
            owap.getObjectWeather().forEach((k, v) -> {
                ow.put(k, deepCloneObjectWeather(v));
            });
            for (Map.Entry<LocalDateTime, ObjectWeather> entry : ow.entrySet()) {
                //        if (entry.getKey().format(DateTimeFormatter.ofPattern("dd.MM.yyyy HH:mm")).equals("29.04.2020
                //        20:00") && rf==0.25)
                //        {
                //        System.out.println(rf);
                //        }
            }
        }
    }
}

```

```

//          if (entry.getKey().format(DateTimeFormatter.ofPattern("dd.MM.yyyy HH:mm")).equals("01.01.2020
10:00")) {
//          System.out.println("3 - " + entry.getKey() + " " + entry.getValue().getAverageSigma_t());
//          }
if (entry.getValue().getPower() / entry.getValue().getStudyForecast() >= (1 / (1 - rf))
    || entry.getValue().getPower() / entry.getValue().getStudyForecast() <= (1 - rf)) {
double numerator = Math.log(entry.getValue().getStudyForecast() / entry.getValue().getPower());
double denominator = 0;
for (int i = 0; i < owap.getSolutions().get(IDX_SIGMA_SOLUTION).size(); i++) {
    numerator    =    numerator    +    owap.getSolutions().get(IDX_SIGMA_SOLUTION).get(i)    *
Math.pow(entry.getValue().getAverageSigma_t(), i + 1);
    denominator = denominator + owap.getSolutions().get(IDX_SIGMA_SOLUTION).get(i) * 3.0 / (i + 3.0);
    // System.out.println(2.0 / (i + 2.0));
}
double sigma = numerator / denominator;
if (sigma <= 0) {
    entry.getValue().setAverageSigma_t(0);
} else if (sigma >= 1) {
    entry.getValue().setAverageSigma_t(1);
}
}
}
final double[] y = new double[countVars * pow + 1];
final double[][] a = new double[countVars * pow + 1][countVars * pow + 1];
QuadraticForecast qf = new QuadraticForecast(configParams);
qf.setY(y, pow, ow);
qf.setA(a, pow, ow);
RealVector solutions = qf.getSolution(a, y);
if (solutions != null) {
    setSolutions(solutions, owap, pow);
    qf.setInfluenceFunctions(owap.getSolutions(), owap.getObjectWeather()); //pow,
    qf.setForecast(owap.getObjectWeather());
    return true;
}
} catch (Exception ex) {
    errorHandler(ex, ex.toString());
}
return false;
}

```

}

### A.1.6 Порівняння прогнозів та динамічна корекція прогнозу

```

public void createForecast(ConfigParams configParams, TreeMap<LocalDateTime,
    ObjectWeather> forecastWeatherData, int messageId) {
    try {
        ArrayList<String> msg = new ArrayList<>();
        ArrayList<ArrayList<String>> reportMsg = new ArrayList<>();
        TreeMap solutionList = new TreeMap();
        TreeMap<Double, TreeMap<Integer, ArrayList<Double>>> solutionsList = new TreeMap<>();
        //isStartStudy = true;
        Path path = Paths.get(configParams.pathSolutionsFile);
        if (isStartStudy && configParams.isRemoveSolutionsFile || !Files.exists(path)) {
            startStudy(configParams, messageId, solutionList, solutionsList, forecastWeatherData.firstKey());
            if (solutionsList.size() > 1) {
                saveSolutions(configParams.pathSolutionsFile, solutionsList.get(solutionsList.lastKey()));
            } else {
                saveSolutions(configParams.pathSolutionsFile, solutionsList.get(solutionsList.firstKey()));
            }
        }
        try (Reader reader = Files.newBufferedReader(path)) {
            WorkWithDB workWithDB = new WorkWithDB(configParams);
            GsonBuilder gsonBuilder = new GsonBuilder();
            Gson gson = gsonBuilder.setPrettyPrinting().create();
            Solutions solutions = gson.fromJson(reader, Solutions.class);
            QuadraticForecast qf = new QuadraticForecast(configParams);
            qf.setWeatherAveragesValues(forecastWeatherData, false);
            setSigma(forecastWeatherData, configParams.parameters.sigmaUp);
            qf.setS_t(forecastWeatherData);
            qf.setAveragesS_t(forecastWeatherData, configParams.parameters.actualPs
configParams.parameters.cp);
            qf.removeNan(forecastWeatherData);
            qf.removeAverageNan(forecastWeatherData);
            qf.removeZeroS(forecastWeatherData);
            qf.setPow(forecastWeatherData, solutions.solutionList.get(1).size());
            //-----
            //setCorrectionClouds(forecastWeatherData);
            //-----
            qf.setInfluenceFunctions(solutions.solutionList, forecastWeatherData); //pow,
            qf.setStudyForecastFromClouds(forecastWeatherData);

```

```

boolean isClassmates = cmd.hasOption("classmates") & configParams.forecast.classmates;
// nearest classmates
if (isClassmates) {
    TreeMap<LocalDateTime, ObjectWeather> nearestNeighbors = new TreeMap<>();
    ZoneId zoneId = ZoneId.of(configParams.parameters.timeZone);
    long from = forecastWeatherData.firstKey().minusYears(1).minusMonths(1).atZone(zoneId).toEpochSecond();
    long to = forecastWeatherData.firstKey().toLocalDate().atStartOfDay().atZone(zoneId).toEpochSecond();
    workWithDB.getNearestNeighborsData(nearestNeighbors, from, to);
    if (nearestNeighbors.size() > 0) {
        LOG.info("nearest neighbors " + nearestNeighbors.firstKey() + " " + nearestNeighbors.lastKey());
        QuadraticForecast qf1 = new QuadraticForecast(configParams);
        qf1.setWeatherAveragesValues(nearestNeighbors, false);
        qf1.setS_t(nearestNeighbors);
        qf1.setAveragesS_t(nearestNeighbors, configParams.parameters.actualPs *
configParams.parameters.cp);
        qf1.removeNan(nearestNeighbors);
        qf1.removeZeroS(nearestNeighbors);
        checkStudyData(configParams.parameters.actualPs, nearestNeighbors);
        Map<LocalDateTime, ObjectWeather> reverseMap
            = nearestNeighbors.descendingMap();
        forecastWeatherData.forEach((k, v) -> {
            getNearestClassmates(v, reverseMap, 2,
                100);
        });
    } else {
        isClassmates = false;
    }
}
setForecastWithCoef(configParams, forecastWeatherData, isClassmates);
if (isClassmates) {
    StringBuilder sb = new StringBuilder();
    forecastWeatherData.forEach((k, v) -> {
        if (k.getHour() > 9 && k.getHour() < 16) {
            double check1 = v.getNearestNeighborsForecast() / v.getForecastWithCoef();
            if (check1 >= 2 || check1 <= 0.5) {
                sb.append("Ошибка прогноза
однокласники\n").append(k.format(DateTimeFormatter.ofPattern("dd.MM.yyyy HH:mm")))
                    .append(String.format(" -> %.2f/%.2f = %.2f\n",

```



```

        v.getNearestNeighborsForecast(), v.getForecastWithCoef(), check1));
    }
    double check2 = Math.abs((v.getAverageSigma_t() - 1) * v.getAverageS_t() /
v.getForecastWithCoef());
    if (check2 >= 3 || check2 <= 1 / 3) {
        sb.append("Ошибка прогноза
облачность\n").append(k.format(DateTimeFormatter.ofPattern("dd.MM.yyyy HH:mm")))
        .append(String.format(" -> (%.2f - 1) * %.2f / %.2f = %.2f\n",
        v.getAverageSigma_t(), v.getAverageS_t(), v.getForecastWithCoef(), check2));
    }
    }
    });
    if (sb.length() > 0) {
        sb.insert(0, configParams.xmlParameters.fileName + "\n");
        dispatcherMessage(new Exception("Ошибка прогнозирования"), sb.toString());
    }
    }
    if (cmd.hasOption("forecastonly")) {
        new CreateOnlyForecastReport().printForecast(configParams, configParams.xmlParameters.fileName +
"_ensemble",
        forecastWeatherData, reportMsg);
        System.exit(0);
    }
    countUpForecastHourErrors(forecastWeatherData);
    countMonthPenaltyImbalance(forecastWeatherData, msg, reportMsg);
    countUpForecastWithCoefHourErrors(forecastWeatherData);
    msg.add("С коэффициентом");
    reportMsg.add(new ArrayList<String>() {
        {
            add("С коэффициентом");
            add("");
            add("");
            add("");
        }
    });
    countMonthPenaltyImbalance(forecastWeatherData, msg, reportMsg);
    msg.forEach(Worker::printItemList);
    if (forecastWeatherData.firstKey().toLocalDate().isEqual(forecastWeatherData.lastKey().toLocalDate())) {
        TreeMap<LocalDateTime, Double> powerList = new TreeMap<>();

```

```

TreeMap<LocalDateTime, Double> updatePowerList = new TreeMap<>();
TreeMap<LocalDateTime, Double> forecast = new TreeMap<>();
forecastWeatherData.forEach((k, v) -> forecast.put(k, v.getStudyForecast()));
ZoneId zoneId = ZoneId.of(configParams.parameters.timeZone);
workWithDB.getLastPowerData(powerList,
    forecastWeatherData.firstKey().toLocalDate().atStartOfDay().atZone(zoneId).toEpochSecond(),
    configParams.parameters.consumption);
if (powerList.size() == 0) {
    LocalDateTime ldt = forecastWeatherData.firstKey().toLocalDate().atStartOfDay();
    LocalDate ld = forecastWeatherData.firstKey().toLocalDate();
    while (ldt.toLocalDate().equals(ld)) {
        powerList.put(ldt.plusHours(1), 0.0);
        ldt = ldt.plusHours(1);
    }
} else if (powerList.get(powerList.firstKey()) != 0 && powerList.get(powerList.lastKey()) == 0 ||
    powerList.get(powerList.firstKey()) == 0 && powerList.get(powerList.lastKey()) != 0) {
    LocalDate ld = powerList.firstKey().toLocalDate();
    powerList.clear();
    workWithDB.getLastPowerData(powerList,
        ld.minusDays(1).atStartOfDay().atZone(zoneId).toEpochSecond(),
        configParams.parameters.consumption);
}
StringBuilder print = new StringBuilder();
powerList.forEach((k, v) -> print.append(k).append(":").append(v).append(" ");
LOG.info(print);
Duration duration = Duration.between(powerList.firstKey(), forecastWeatherData.firstKey());
powerList.forEach((k, v) -> updatePowerList.put(k.plusDays(duration.toDays()), v));
updatePowerList.forEach((k, v) -> {
    ArrayList<Boolean> consumption = new ArrayList<>();
    consumption.add(true);
    double coef = getTempCoef(configParams, k, consumption);
    if (forecastWeatherData.containsKey(k)
        && coef > 0) {
        double forecastPower = v;
        if (!k.isEqual(forecastWeatherData.firstKey()) && !k.isEqual(forecastWeatherData.lastKey())
            || (k.isEqual(forecastWeatherData.firstKey()) || k.isEqual(forecastWeatherData.lastKey())
                && v > 0)) {
            forecastPower = round(forecastWeatherData.get(k).getForecastWithCoef() * coef, 2, k);
            if (forecastPower > configParams.parameters.actualPs) {

```

```

        forecastPower = configParams.parameters.actualPs;
    }
}
updatePowerList.replace(k, forecastPower);
} else {
    if (!consumption.get(0) && coef == 0) {
        updatePowerList.replace(k, 0.0);
    } else if (coef == 0 && v > 0) {
        updatePowerList.replace(k, 0.0);
    }
}
});
print.setLength(0);
updatePowerList.forEach((k, v) -> print.append(k).append(":").append(v));
LOG.info(print);
final double[] sumPower = {0};
updatePowerList.forEach((k, v) -> {
    sumPower[0] = sumPower[0] + v;
    if (forecastWeatherData.containsKey(k)) {
        LOG.info(k.format(DateTimeFormatter.ofPattern("dd.MM.yyyy HH:mm"))) + " : "
            + round(forecastWeatherData.get(k).getStudyForecast(), 1) + " -> "
            + round(forecastWeatherData.get(k).getForecastWithCoef(), 1) + " -> "
            + round(v, 1));
    } else {
        LOG.info(k.format(DateTimeFormatter.ofPattern("dd.MM.yyyy HH:mm"))) + " : *** -> *** -> " +
round(v, 1));
    }
});
LOG.info(sumPower[0]);
if (messageId != 0) {
    telegramBot.editMsg(mainConfigParams.telegram.forecastChannel, messageId, "подготовка отчета
100%");
}
if (isClassmates) {
//            new CreateReport().printForecast(configParams, configParams.xmlParameters.fileName +
"_neighbor",
            new CreateReport().printForecast(configParams, configParams.xmlParameters.fileName +
"_ensemble",
            excelReportMailList,

```

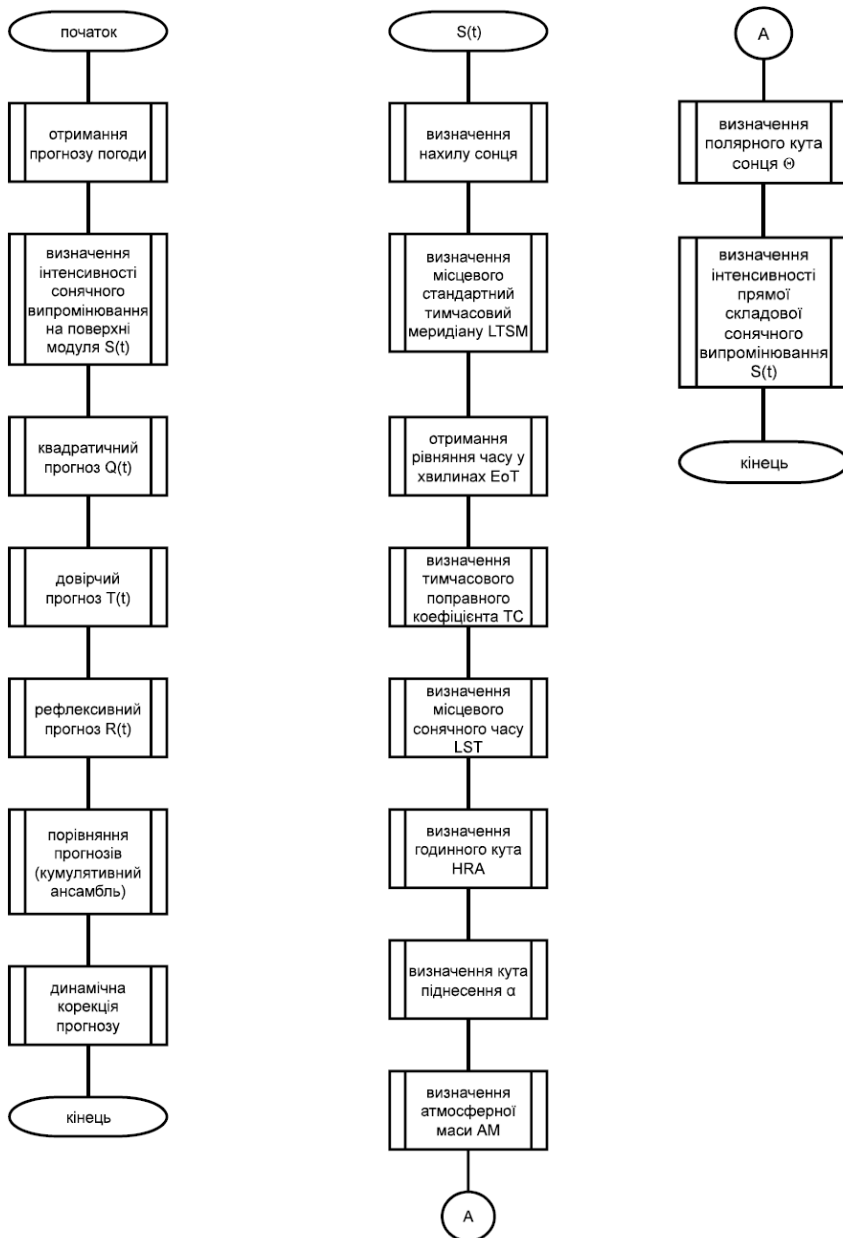
```

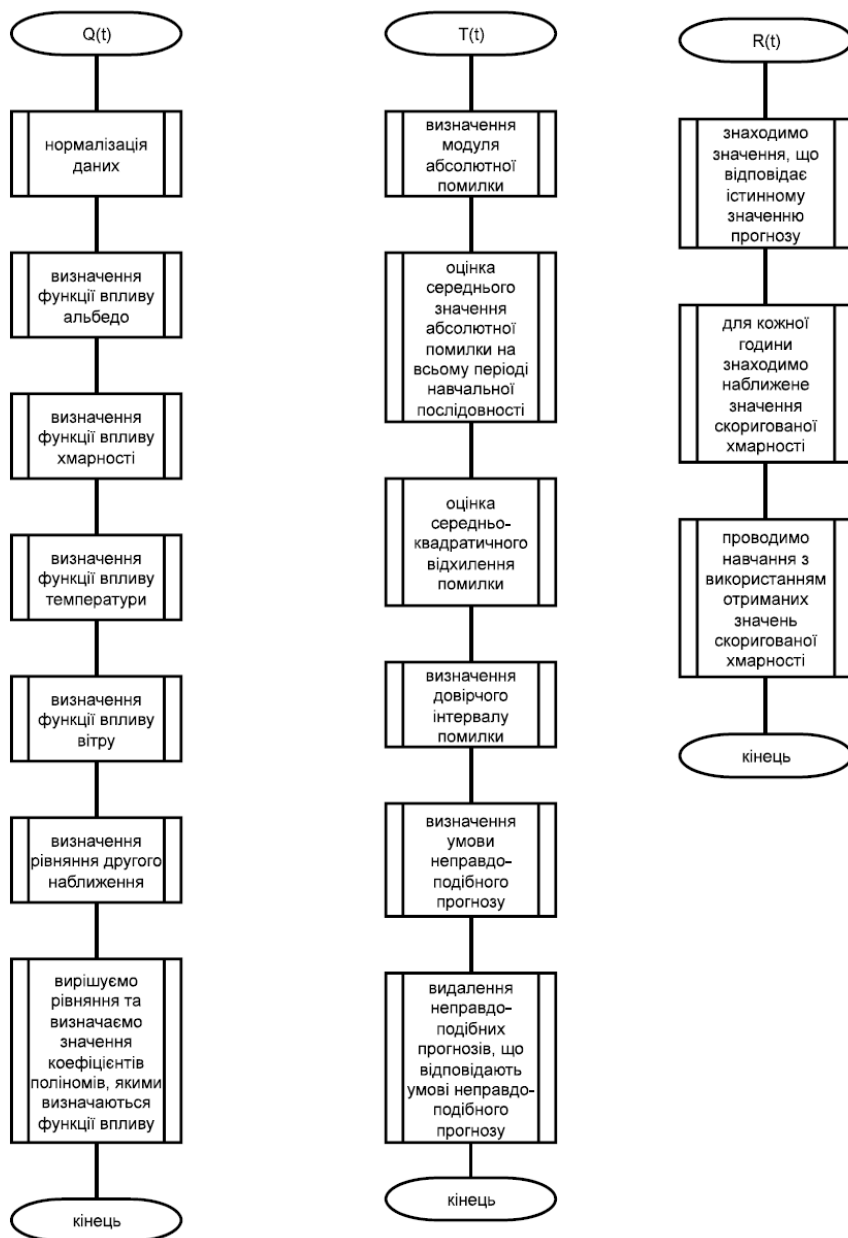
        forecastWeatherData, reportMsg, updatePowerList);
    } else {
        new    CreateReport().printForecast(configParams,    configParams.xmlParameters.fileName    +
        "_forecast",
            excelReportMailList,
            forecastWeatherData, reportMsg, updatePowerList);
    }
//    temp
    setDataToDataBase(workWithDB, forecast, solutions);
    createXMLReport(configParams, updatePowerList);
// end temp
} else {
    if (isClassmates) {
        new    CreateReport().printForecast(configParams,    configParams.xmlParameters.fileName    +
        "_ensemble",
            excelReportMailList,
            forecastWeatherData, reportMsg, null);
    } else {
        new    CreateReport().printForecast(configParams,    configParams.xmlParameters.fileName    +
        "_forecast",
            excelReportMailList,
            forecastWeatherData, reportMsg, null);
    }
}
}
} catch (Exception ex) {
    errorHandler(ex, ex.toString());
}
}
}

```

## Додаток Б

### Алгоритм роботи програми





Ім'я користувача:  
Олег Бойко

ID перевірки:  
1013276117

Дата перевірки:  
12.12.2022 12:02:50 EET

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
12.12.2022 12:03:19 EET

ID користувача:  
100008838

Назва документа: 01\_151м-21-1\_Чорномор\_Р\_Р\_-\_ПЗС

Кількість сторінок: 114 Кількість слів: 18010 Кількість символів: 134145 Розмір файлу: 2.35 MB ID файлу: 1013034975

## 12.6% Схожість

Найбільша схожість: 3.31% з Інтернет-джерелом (<https://avenston.com/articles/solar-pv-forecast>)

11% Джерела з Інтернету 422 ..... Сторінка 116

1.85% Джерела з Бібліотеки 174 ..... Сторінка 119

## 0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

## 0% Вилучень

Немає вилучених джерел

## Модифікації

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

Замінені символи 42