

Довідка
про відсутність плагіату у випускній кваліфікаційній роботі

Міністерство освіти і науки України
Український державний університет науки та технологій

Кафедра «Комп'ютерні інформаційні технології»

ДОВІДКА

За результатами перевірки випускної кваліфікаційної роботи здобувача вищої освіти
Шановала Данила Олеговича

на тему: Дослідження методів і процедур класифікації та прогнозування
недетермінованих процесів на основі показників хаотичної динаміки

в роботі не виявлено порушень академічної доброчесності.

Керівник ВКР Беканюк /Владислав Шановський/

20.12.2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Український державний університет науки і технологій

Кафедра Комп'ютерні інформаційні технології

«ДО ЗАХИСТУ»

Завідувач кафедри

В /Вадим ГОРЯЧКІН/

« 12 » 12 20 21 р.

ДИПЛОМНА РОБОТА

на здобуття освітнього ступеня «магістр»

Галузь знань **12 Інформаційні технології**

Спеціальність **121 Інженерія програмного забезпечення**

Тема **Дослідження методів і процедур класифікації та прогнозування недетермінованих процесів на основі показників хаотичної динаміки**

Керівник дипломної роботи

проф. Владислав СКАЛОЗУБ

Нормоконтролер

доц. Олена КУРОП'ЯТНИК

Студент групи ПЗ2021

Данило ШАПОВАЛ

Student

Danylo SHAPOVAL

Дніпро – 2021

Дніпропетровський національний університет залізничного транспорта

Факультет Комп'ютерні технології і системи кафедра Комп'ютерні інформаційні технології

Спеціальність Інженерія програмного забезпечення

«ЗАТВЕРДЖУЮ»

Завідувач кафедри


(підпис)

«__» _____ 2021 р.

ЗАВДАННЯ

до дипломного проекту на здобуття ОС Магістр

(освітньо-кваліфікаційний рівень)

студента групи ПЗ2021 Шаповала Данила Олеговича

(номер групи)

(ПІБ)

1 Тема дипломного проекту: Дослідження методів і процедур класифікації а прогнозування недетермінованих процесів на основі показників хаотичної динаміки

затверджена наказом по університету від «12» листопада 2020 р. № 8.

2 Термін подання студентом закінченого проекту «12» грудня 2021 р.

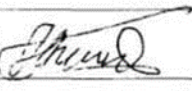
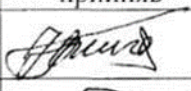


3 Вихідні дані до дипломного проекту

часові послідовності спостережень недетермінованих технологіко-економічних процесів сфери функціонування залізничного транспорту України, методи та процедури дослідження властивостей часових послідовностей даних, попередні результати класифікації числових послідовностей.

4 Зміст пояснювальної записки (перелік питань до розробки) аналіз та дослідження методів та процедур моделювання недетермінованих антиперсистентних часових послідовностей та її прогнозування, аналіз та дослідження порівняльної ефективності алгоритмів агрегування та інтерполяції при моделюванні антиперсистентних часових послідовностей, розробка програмних інструментальних засобів, проведення експериментів із порівняльного аналізу ефективності алгоритмів агрегування елементів часових послідовностей, охорона праці та безпеки в надзвичайних ситуаціях, технічна документація на розроблені програмні засоби.

5 Перелік демонстраційного матеріалу презентація завдання дослідження методів та процедур моделювання недетермінованих антиперсистентних часових послідовностей, процедури порівняльного аналізу алгоритмів моделювання, результати та висновки щодо ефективності алгоритмів аналізу та прогнозування, розроблений програмний комплекс та інструментальні заходи аналізу, відео-демонстрація роботи розробленого програмного комплексу для проведення дослідження

6. Консультанти (з назвами розділів):

Розділ	Консультант	Підпис, дата	
		завдання видав	завдання прийняв
Кошторис на розробку ПЗ	доцент Гненний М.В.		
Охорона праці та безпека в надзвичайних ситуаціях	Професор Саблін О. І.		

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва розділів дипломної роботи	Термін виконання розділів роботи	Примітка
1	Вступ	10.11.2021-15.11.2021	5%
2	Аналіз сучасного стану дослідження проблеми класифікації ЧР за науковими джерелами	15.11.2021-22.11.2021	10%
3	Аналіз засобів та сучасного стану програмно-апаратного забезпечення, яке потребує вдосконалення для вирішення завдань дослідження	22.11.2021-22.11.2021	20%
4	Постановка задачі дослідження, складання технічного завдання	23.11.2021-30.11.2021	30%
5	Техніко-економічні показники	01.12.2021-07.12.2021	40%
6	Розробка моделей, алгоритмів та інструментальних засобів дослідження	08.12.2021-12.12.2021	50%
7	Проведення досліджень	13.12.2021-15.12.2021	60%
8	Оформлення тез доповідей	15.12.2021-22.12.2021	80%
9	Підготовка статті у фаховий журнал	15.12.2021-22.12.2021	90%
10	Оформлення пояснювальної записки	15.12.2021-22.12.2021	100%
11	Розробка демонстраційних матеріалів	15.12.2021-22.12.2021	100%

Дата видачі завдання «18» 09 2020 р.

Керівник дипломного проєкту Веканит /Скалозуб В. В. /
(підпис) (ПІБ)

Завдання прийняв до виконання ШП /Шаповал Д.О. /
(підпис) (ПІБ)

РЕФЕРАТ

Об'єктом дослідження є процеси моделювання та дослідження параметрів недетермінованих часових послідовностей з рівномірним, які являються антиперсистентними часовими рядами.

Предметом дослідження є процедури аналізу та класифікації моделей антиперсистентних часових послідовностей з рівномірним кроком між рівнями спостережень,

Метою поданої роботи є підвищення ефективності та точності методів та алгоритмів класифікації, моделювання та прогнозування антиперсистентних часових послідовностей спостережень.

Методи дослідження: методи аналізу часових послідовностей, математичної статистики, методи порівняльного аналізу, комп'ютерне моделювання, процедури короткострокового прогнозування. Методи проектування та створення програмного забезпечення.

Результати та їх новизна: виконано порівняльний аналіз чисельної ефективності алгоритмів класифікації та короткострокового прогнозування антиперсистентних часових послідовностей спостережень, удосконалено процедуру дослідження зазначених процесів.

Розрахунково-пояснювальна записка складається зі вступу, 5 розділів, висновків, бібліографічного списку та додатки

Вступ – містить сутність роботи та актуальність досліджень та розробок (2 сторінки).

Перший розділ – опис результатів аналізу сучасного стану досліджень завдань моделювання недетермінованих, зокрема, антиперсистентних процесів за науковими літературними джерелами (8 сторінок).

Другий розділ – представлено сутність та відмінність досліджень алгоритмів антиперсистентних процесів, а також структуру процедур інтерполяції та прогнозування на основі агрегування даних спостережень (7 сторінок).

Третій розділ – представляє структуру, процес проектування і розробки інструментального забезпечення, призначеного для проведення дослідження (14 сторінок).

Четвертий розділ – дослідження властивостей методу та їх результати (24 сторінок).

П'ятий розділ – розкриті питання охорони та безпеки праці в надзвичайних ситуаціях (10 сторінок).

висновки Складаються з 1 сторінки;

список літератури – включає в себе список використаної літератури. Складає 5 сторінки;

Додатки – технічне завдання і робочий проект. Таблиць – 6, рисунків – 59, бібліографія – 70.

Ключові слова: антиперсистентна часова послідовність, класифікація моделей, ефективність алгоритмів, короткострокове прогнозування, порівняльне дослідження.

ЗМІСТ

Вступ.....	8
1. Постановка завдань дослідження, аналіз методів та засобів моделювання недетермінованих часових послідовностей та її прогнозування	12
1.1. Аналіз предметної сфери моделювання недетермінованих процесів	12
1.1.1. Опис особливостей завдань моделювання. Актуальність дослідження	12
1.1.2. Огляд результатів основних наукових досліджень	16
1.1.3. Огляд засобів програмних аналогів	22
1.2. Призначення та сфера застосування результатів досліджень	26
1.3. Постановка завдань дослідження	26
Висновки до розділу 1	27
2. Обґрунтування та методика досліджень алгоритмів із моделювання та прогнозування антиперсистентних процесів,.....	28
2.1. Дослідження ефективності алгоритмів із моделювання та прогнозування антиперсистентних процесів,.....	28
2.1.1. Процедури аналізу та класифікації антиперсистентних процесів	28
2.1.2. Аналіз напрямків удосконалення алгоритмів класифікації та короткострокового прогнозування антиперсистентних процесів	34
2.1.3. Методика та процедури дослідження чисельної ефективності та точності алгоритмів класифікації та інтерполяції даних антиперсистентних процесів . Порівняльний аналіз.	36
2.2. Дослідження функціональної ефективності алгоритмів моделювання ...	39
2.3. Особливості структури процесів моделювання антиперсистентних числових послідовностей	43
Висновок до розділу 2	46
3. Розробка структури та інструментальних засобів для дослідження алгоритмів моделювання недетермінованих часових послідовностей	47
3.1. Формалізація завдань дослідження алгоритмів моделювання.....	47
3.2. Розробка базової архітектури інструментальних засобів	48
3.3. Внутрішнє проектування інструментальних засобів	49

3.3.1. Обґрунтування та вибір мови програмування	49
3.3.2. Розробка архітектури ієрархії та взаємодії класів системи інструментальних засобів	50
3.3.3. Застосування принципів проектування архітектури програмних систем	50
3.4. Формування інтерфейсу користувача програмного комплексу	53
3.5. Тестування та налагодження програми	53
3.5.1. Аналіз та застосування методів тестування та налагодження	55
Висновки до розділу 3	63
4. Порівняльні дослідження ефективності алгоритмів класифікації та моделювання антиперсистентних числових послідовностей	64
4.1. Постановка завдань та підготовка даних експерименту	64
4.1.1. Опис структури засобів програмно-апаратного середовища	64
4.1.2. Алгоритми та процедури методу класифікації та моделювання антиперсистентних часових послідовностей	64
4.1.3. Опис методики відбору недетермінованих послідовностей для експериментальних досліджень алгоритмів	66
4.2. Процедура та результати проведення експериментальних досліджень алгоритмів моделювання антиперсистентних часових послідовностей	68
Висновки до розділу 4	75
5. Охорона праці та безпека в надзвичайних ситуаціях	76
5.1. Загальні положення	76
5.2. Шкідливі виробничі фактори на робочому місці на підприємстві	79
5.2.1. Мікроклімат	81
5.2.2. Освітлення робочого місця	82
5.2.3. Шум і вібрація	82
5.3. Дії працівників в надзвичайній ситуації	83
Висновки	86
Список використаної літератури	88
Додатки	96

ВСТУП

Актуальність роботи. Комп'ютерне моделювання натеper являється одним із основних засобів аналізу та прогнозування складних процесів у багатьох сферах діяльності. Моделі, методи та засоби щодо коректного та достовірного застосування процедур комп'ютерне моделювання були і залишаються актуальними. Складність процесів дослідження являється головним із факторів, які визначають можливості та ефективність процедур аналізу та прогнозування сучасних систем, с тому числі процесів функціонування залізничного транспорту. На підставі даних та результатів моделювання вирішується надзвичайно широкий спектр завдань щодо розрахунку поточних параметрів складних систем, оцінки можливих станів систем, прогнозування очікуваних наслідків управління, результатів вибору варіантів оптимальних рішень тощо.

При аналізі та моделюванні складних систем в багатьох випадках в якості вихідних даних використовують часові послідовності процесів, які представляють показники досліджуваних величин. Формування та інтерпретація таких послідовностей даних потребує в тому числі коректного застосування сукупності математико-статистичних методів аналізу. Також ці методи вимагають досить значну кількість даних, виконання умов певних припущень, наприклад, постулати нормальності ін. Серед важливих вимог до формування та застосування методів комп'ютерне моделювання визначається наявність трендів досліджуваних процесів. Розвиток методів та засобів хаотичної динаміки дозволив встановити вимоги, які по-суті забезпечують можливість застосування процедур моделювання та прогнозування процесів, представлених часовими послідовностями, часовими рядами (ЧР). Таким показником являється параметр Херста, за яким воконуюється класифікація процесів на персистентні (трендостійкі), антиперсистентні (для них передбачається лише можливість зміни тенденції розвитку), хаотичні, При цьому методи аналізу, комп'ютерного

моделювання та прогнозування антиперсистентні часових послідовностей досліджені не в повній мірі.

При моделюванні недетермінованих зокрема антиперсистентних часових послідовностей даних, які представляють складні процеси різної природи, виникають питання щодо визначення можливості коректного формування комп'ютерних моделей, процедур визначення параметрів станів процесів, методики дослідження властивостей часових послідовностей, точність отриманої моделі тощо.

Розвиток інструментарію та програмних засобів моделювання антиперсистентних часових послідовностей розширює сферу застосування програмної інженерії. Розвиток досліджень щодо моделей, алгоритмів і програмних засобів, призначених для аналізу, класифікації та прогнозування антиперсистентних часових послідовностей на тепер мають науковий і практичний інтерес, що свідчить про актуальність представленої роботи.

Методи аналізу та дослідження, що використовувалися в роботі, містять методи системного аналізу, методи аналізу часових послідовностей, методи математичного моделювання та математичної статистики, методи порівняльного аналізу, комп'ютерне моделювання, процедури короткострокового прогнозування. Методи проектування та створення програмного забезпечення, а також методи алгоритмізації; експериментального дослідження властивостей програм.

Об'єктом дослідження є процеси моделювання та дослідження параметрів недетермінованих часових послідовностей з рівномірним, які являються антиперсистентними часовими рядами.

Предметом дослідження являються чисельні процедури аналізу, класифікації та прогнозування моделей антиперсистентних часових послідовностей з рівномірним кроком між рівнями спостережень,

Мета роботи це підвищення ефективності та точності методів та алгоритмів класифікації, моделювання та прогнозування антиперсистентних часових послідовностей даних спостережень.

Завдання. Завдання дослідження полягали у наступному:

- розробити алгоритмічні та програмні засоби та виконати порівняльний аналіз чисельної ефективності алгоритмів класифікації антиперсистентних часових послідовностей спостережень
- розробити алгоритмічні та програмні засоби та виконати короткострокове прогнозування антиперсистентних часових послідовностей,
- розробити інструментальне програмне середовище, яке забезпечує коректне дослідження алгоритмів моделювання антиперсистентних часових послідовностей;
- виконати порівняльний аналіз алгоритмів моделювання антиперсистентних часових рядів щодо точності та чисельної ефективності;
- отримати рекомендації стосовно процедур моделювання антиперсистентних часових послідовностей.

Методи дослідження. Для дослідження ефективності алгоритмів моделювання недетермінованих процесів були використані методи аналізу часових послідовностей, методи математичної статистики, системного аналізу, методи аналізу часових послідовностей, методи математичного моделювання та математичної статистики, методи порівняльного аналізу, комп'ютерне моделювання, процедури короткострокового прогнозування. Методи проектування та створення програмного забезпечення, а також методи алгоритмізації; методи експериментального дослідження властивостей програм.

Наукова новизна результатів роботи визначається наступним. В роботі виконано порівняльний аналіз чисельної ефективності алгоритмів класифікації та короткострокового прогнозування антиперсистентних часових послідовностей спостережень, удосконалено процедуру дослідження

зазначених процесів, забезпечує коректність результатів комп'ютерного моделювання та підвищення точності прогнозування показників процесів.

Практичне значення мають результати проведеного аналізу стану застосування методів та засобів комп'ютерного моделювання антиперсистентних часових послідовностей спостережень, розробки та порівняльного аналізу алгоритмів моделювання, результати порівняльного аналізу чисельної ефективності алгоритмів класифікації антиперсистентних часових послідовностей спостережень, рекомендації стосовно процедур моделювання антиперсистентних часових послідовностей, а також інструментальні програмні засоби аналізу антиперсистентних процесів. Дослідження завдань щодо порівняльного аналізу алгоритмів класифікації та моделювання антиперсистентних часових послідовностей автору не відомі.

Апробація результатів дослідження. Процес та результати дослідницької роботи доповідались на семінарах кафедри КІТ 08.12.2021 р. Результати порівняльного аналізу алгоритмів класифікації та моделювання антиперсистентних часових послідовностей доповідалися на 15 міжнародній науково-практичній конференції «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті» (16.12.2021 р.) м. Дніпро, Державний університет науки і технологій.

1. ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ, АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ МОДЕЛЮВАННЯ НЕДЕТЕРМІНОВАНИХ ЧАСОВИХ ПОСЛІДОВНОСТЕЙ ТА ЇЇ ПРОГНОЗУВАННЯ

1.1. Аналіз предметної сфери моделювання недетермінованих процесів предметної сфери.

1.1.1. Опис особливостей завдань моделювання. Актуальність дослідження .

Недетерміновані часові послідовності (НЧП) представляють надзвичайно широкий спектр складних технологічних, економічних, інформаційних, соціальних та інших процесів. Їх отримання, аналіз та математичне і комп'ютерне моделювання має значний прикладний та науковий інтерес, широке практичне застосування. При цьому відзначається що такі завдання мають багато проблем щодо повноти та достовірності даних, коректності та адекватності моделей і методів та інших. Серед різноманітних завдань аналізу і використання моделей НЧП відзначаються аналіз (як розрахунок певних характеристик процесів), моделювання (створення певних форм, за допомогою яких можливо замінити процес алгоритмом або «засобами оцінювання властивостей» з визначеною точністю), прогнозування (отримання оцінок майбутніх значень характеристик НДП). Розрізняють оперативне, короткострокове, довгострокове та інші форми прогнозування. Важливими НДП являється обставини пов'язані з факторами невизначеності, під дією яких формуються самі процеси. Це суттєво ускладнює аналіз та моделювання таких процесів, представлених часовими рядами (ЧР) (послідовностями) результатів спостережень.

Натепер все більше актуальною стає задача аналізу, прогнозування та оперативного планування значень параметрів економічних і технологічних процесів складних систем, до яких відносяться і процеси залізничного транспорт, в умовах змін зовнішнього середовища. Вони виникають на

різних рівнях прийняття рішень, а їх достовірне та вчасне вирішення забезпечує ефективне функціонування, як окремого підрозділу (підсистеми), так і всього підприємства (досліджуваної системи) в цілому.

Переважна більшість ЧР, що описують технологіко-економічні процеси, є одновимірними, характеризують їх лише одним значенням. Дослідження побудовані на основі ЧР спираються лише на значення показника (параметр, властивість), який характеризує цей ряд. Через це відзначається бурхливий розвиток методів їх дослідження та аналізу.

На рис.1.1 наведена класифікація часових рядів. Відповідно рис. 1.1 розрізняють наступні види ЧР повні (періоди часу між рівнями ряду рівні), неповні (періоди часу між рівнями ряду нерівні), одновірні (характеризують зміну тільки одного явища/показника), багатомірні (характеризують зміну декількох явищ/показників), інтервальні (рівні ряду характеризують величину явища/показника за певні періоди часу - година, день, місяць, та ін.), моментні (рівні ряду характеризують величину явища/показника за певні моменти часу – початок тижня, місяця, кварталу та ін.), персистентні (або як такі що мають «пам'ять», ряд, що збереже свою поточну тенденцію в найближчому майбутньому), антиперсистентні (ряд, що змінить свою поточну тенденцію в найближчому майбутньому).

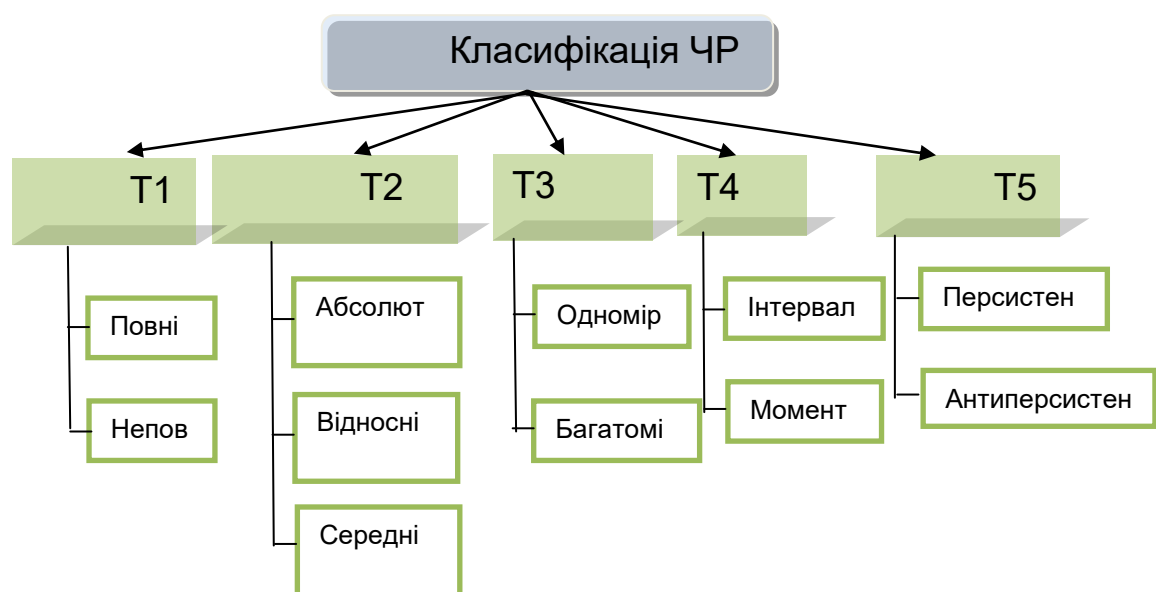


Рисунок 1.1. Класифікація типів часових рядів

Класифікація рис. 1.1 типів ЧР демонструє широкий спектр можливих типів та їх способів представлень. Класи утворені відповідно наступних ознак: T1 – за повнотою (повні та неповні), T2 – за способом представлення даних (абсолютні величини, відносні та середні значення показників процесів), T3 – кількість параметрів вимірів показників (одномірні та багатомірні), T4 – представлення часу (інтервальні та моментні), T5 – за типом процесів (персистентні або трендостійкі, антиперсистентні). Відзначимо саме клас T5, який містить антиперсистентні процеси, дослідження алгоритмів моделювання котрих виконано в цій роботі.

В природі та у технологіях існує велика кількість ЧР, рівні яких не розподілені за гауссовським законом розподілу. Зазначимо, що при цьому і процеси залізничного транспорту не є виключенням. Г. Херст досліджуючи природу розливу Нілу, з'ясував що динаміка розливів Нілу не підпорядковується гіпотезі випадковості та незалежності рівнів ЧР. Його подальші дослідження привели до появи R/S-аналізу ЧР. Вагомий внесок в дослідження природи ЧР та розвиток теорії R/S-аналізу вніс Б. Мандельброт. На основі досліджень статистичних даних фінансових бірж він з'ясував, що деякі ЧР не є випадковими, а їх динаміка не випадкова, а підпорядковується деякому степеневому закону та мають довготривалу пам'ять. В подальших своїх дослідженнях Б. Мандельброт заснував теорію фракталів та довів фрактальну природу ЧР фінансових ринків. Як показали дослідження, ЧР багатьох фінансово-економічних показників (валютних курсів, курсів акцій) мають фрактальну структуру, для їх дослідження можна використовувати апарат фрактального аналізу, зокрема R/S аналіз, який базується на обчисленні статистики Херста, що є мірою випадковості часового ряду. У статтях вперше було встановлено, що існують технолого-економічні процеси ЗТ, які представляються антиперсистентними часовими рядами. Питання аналізу, класифікації та прогнозування таких процесів досліджуються у дисертації.

Натепер існує класифікація недетермінованих процесів, представлених ЧР, яка використовує величини константи Херста (H). А саме, при $H > 0,5$ процес є трендостійким, при $H = 0,5$ процес повністю хаотичний, при $H < 0,5$ процес антиперсистентний. Стосовно антиперсистентних процесів встановлюється що вони мають властивість щодо переміни тенденції. Інші властивості для таких ЧР не визначаються, а значить і не використовуються при аналізі та прогнозування. У роботі визначені можливості із класифікації та прогнозування антиперсистентних ЧР, отриманих для процесів залізничного транспорту (ЗТ) України.

У зв'язку зі складністю і недетермінованістю численних процесів залізничного транспорту. одним із основних способів їх опису є представлення даних у вигляді упорядкованих по етапах та інтервалах значень деяких основних вимірюваних характеристик - формування часових рядів (ЧР). Прикладами таких ЧР для залізниці є параметри вагонопотоків різних категорій на полігонах і станціях, оцінки витрат різних видів ресурсів за періодами, параметри вантажопотоків, послідовності подій процесів експлуатації та ін. Моделі і методи аналізу та прогнозування станів і параметрів складних систем на основі ЧР отримали визнання і широке поширення, знаходять нові області застосування і форми реалізації. Разом з тим ЧР процесів залізничного транспорту, а також вимоги до подання результатів їх аналізу, мають певну і важливу специфіку, яка відрізняє їх від аналізу ЧР природних процесів, наприклад, сонячної та атмосферної активності, кардіологічних, діяльності мозку і т. п. Зазначимо тут основні з них: по-перше, ЧР є «короткими»; по-друге, вимога результативності, тобто практичної «придатності» процедур для реалізації достовірного виробничо-технічного прогнозування і планування основних показників діяльності, обумовлених ЧР. Основна увага приділяється процесам діяльності залізничного транспорту, представленим антиперсистентними ЧР вантажних вагонопотоків (ЧР-В).

Численні процеси залізничного транспорту можуть бути представлені і досліджені як ЧР. Встановлено методами фрактального аналізу, що деякі ВР-В є антиперсистентними, при цьому за обсягом даних ЧР є «короткими». Представлена спеціалізована процедура аналізу таких ЧР-В, заснована на класифікації щодо узагальнення послідовних рівнів. За результатами застосований цієї процедури визначається певний «клас» ЧР-В, на основі чого встановлюється об'ємний показник, що характеризує можливості побудови тренду процесу ЧР. Як приклади ЧР-В використано параметри вагонопотоків, сформованих для перевезення зерна в порти України.

Суттєво що ЧР аналізують за даними спостережень без додаткової інформації, без аналізу впливу зовнішніх факторів. Такий аналіз є неповним, але досить часто його достатньо для побудови достовірних моделей процесів. В роботах зазначається, що всі ЧР можуть бути придатні для побудови прогнозу у відповідності до підходів теорії математичної статистики. Такі висновки витікають з необхідності існування для більшість статистичних методів гіпотези щодо деякої закономірності (тренд, сезонність, циклічність), або відповідності ЧР нормальному закону розподілу. Зазначені вимоги до ЧР повинні чітко виконуватися, в іншому випадку існує висока доля ймовірності отримати хибний результат.

1.1.2. Огляд останніх досліджень

Прогнозування займає важливе місце в процесах повсякденного життя і діяльності. Традиційні статистичні підходи до дослідження змін у часі можуть передбачити явища, які виникають у зв'язку з певними новими тенденціями, але не завжди дозволяють прогнозувати дані з допомогою лінгвістичних фактів. Крім того, традиційні ЧР вимагають більше історичних даних (період спостережень над процесом), а також використання деяких припущень, наприклад типу постулатів нормальності.

Різноманітні і численні процеси різної природи і походження знаходять своє відображення у формі недетермінованих, випадкових, нечітких та ін. ЧР.

Дослідженням природи, властивостей ЧР, а також моделям і методам їх аналізу і прогнозування присвячені численні роботи. Навіть при наявності достатньої кількості статистичних даних, застосування стандартних статистичних методів аналізу ЧР не дає можливості побудувати адекватну та достовірну модель.

Дослідження властивостей ЧР деяких процесів перевезення та обслуговування вагонопотоків залізничного транспорту методами фрактального аналізу дозволили класифікувати їх як антиперсистентні. Застосування до таких процесів відомих методик оцінки властивостей, аналізу та планування параметрів призводить до суттєвих додаткових витрат. У той же час відзначається відсутність загальновизнаних і прийнятних для практики методів і процедур, необхідних для аналізу і прогнозування антиперсистентних ЧР. Зазначені вище фактори і стали джерелом і предметом подальшого дослідження.

Процес перевезення вантажів будь-яким видом транспорту представляє собою нерівномірний у часі та об'ємі технологічний процес (рис. 1..2, 1.3).



Рисунок. 1.2. Динаміка відправлення вагонів зі станції протягом календарного року

Представлені ЧР мають складну хаотичну структуру, яка відображатиме особливості технолого-економічних процесів залізничних перевезень. Виявлення загальних властивостей і оцінка параметрів таких ЧР-В дає можливість обґрунтованого планування процесів як організації, так і

виконання залізничних перевезень, прогнозування оцінок кількісних характеристик показників витрат на переробку вагонопотоків та ін.

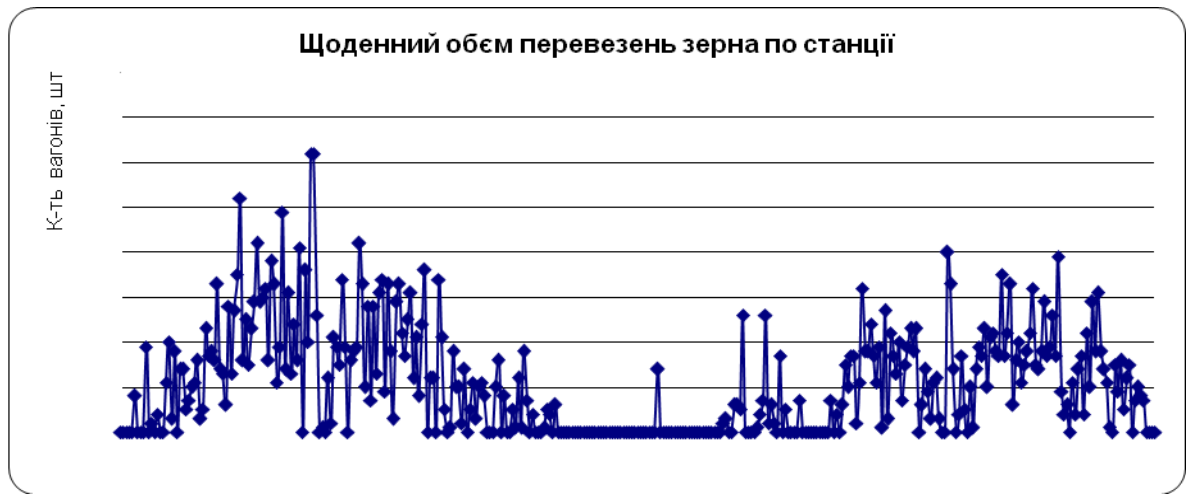


Рисунок. 1.3. Динаміка процесів перевезень зерна

В останні роки особлива увага приділяється методам хаотичної динаміки, фрактального аналізу. На основі цих методів виконані дослідження широкого кола фінансово-економічних, природних і технологічних процесів в складних системах. Дослідження властивостей ЧР деяких процесів перевезення та обслуговування вагонопотоків залізничного транспорту методами фрактального аналізу дозволили класифікувати їх як антиперсистентні. Застосування до таких процесів відомих методик оцінки властивостей, аналізу та планування параметрів призводить до суттєвих додаткових витрат. У той же час відзначається відсутність загальновизнаних і прийнятних для практики методів і процедур, необхідних для аналізу і прогнозування антиперсистентних ЧР. Зазначені вище фактори і стали джерелом і предметом подальшого дослідження..



Рисунок 1.4. Динаміка перевезень зерна (в вагонах) по станції

Представлені ЧР мають складну хаотичну структуру, яка відображатиме особливості технолого-економічних процесів залізничних перевезень. Виявлення загальних властивостей і оцінка параметрів таких ЧР-В дає можливість обґрунтованого планування процесів як організації, так і виконання залізничних перевезень, прогнозування оцінок кількісних характеристик показників витрат на переробку вагонопотоків та ін.

Виконані дослідження ЧР-В, що характеризують вагонопотік на деяких залізничних полігонах, показали, що частина з них мають антиперсистентний характер. Розроблені методи аналізу таких ЧР-В не дозволяє отримати прийнятні для потреб практики методики аналізу, планування і управління. Зокрема, розрахувати достовірні оцінки очікуваних обсягів переробки вагонів на черговому етапі, а також встановити значення пов'язаних з цим витрат різних типів ресурсів.

Для дослідження загальних властивостей ЧР-В використані методи фрактального аналізу, обчислення показника Херста. Його значення дозволяють встановити категорію, класифікувати процеси ЧР на персистентні (трендостійкі), антиперсистентні (злам тенденції, повернення), виявити випадковий характер процесу.

$$H = \frac{\log\left(\frac{R}{S}\right)}{\log(a * N)} \quad (1.1)$$

де H – показник Херста;

S – середнє квадратичне відхилення ряду спостережень;

R – розмах накопиченого відхилення;

N – число періодів спостережень;

a – задана константа.

Було показано, що для «коротких» ЧР (відповідають реальним процесам реалізації залізничних перевезень) краще використовувати величину константи $a = \pi / 2$. Таке значення параметра (1.1) дозволяє з більшою вірогідністю оцінювати властивості ЧР і стверджувати, коли ЧР має пам'ять ($H > 0,5$).

Для аналізу та подальшого дослідження технолого-економічних процесів, що описують ЧР, для яких показник Херста перебуває в області так званого "білого шуму" ($H = [0,4; 0,5]$), запропоновано використати процедуру агрегування рівнів ЧР. Агрегування – об'єднання, укрупнення економічних показників (інформації) за певними ознаками.

В подальшому дослідженні запропонована спеціалізована процедура аналізу ЧР, що дозволяє, по-перше, диференціювати антиперсистентні ЧР, віднести їх до деяких різних класів (що розрізняються числом об'єднаних поряд стоячих рівнів, довжинами інтервалів достовірного планування), по-друге, виконувати процедури прогнозування значень сумарних об'ємних показників модифікованого ряду за встановлений інтервал (період узагальнення даних). Таким чином, в рамках запропонованої процедури відкривається можливість кількісного аналізу зазначених антиперсистентних ЧР-В, але лише на виявленому при класифікації інтервалі.

Разом з цим для практики інтерпретації та планування процесів, поданих ЧР, бажано встановити їх деякі додаткові характеристики, що дають можливість також диференціювати ЧР між собою, встановлювати узагальнені кількісні показники. Отримання подальших рекомендацій із аналізу та планування антиперсистентних ЧР (АЧР) з використанням запропонованої процедури узагальнення рівнів та перетворення вихідного ЧР

(ПУРАЧР), з подальшим дослідженням його властивостей на основі моделі параметрів Херста(1.1).

На основі процедури ПУРАЧР вирішуються такі ключові питання.

1. Класифікація антиперсистентних ЧР.
2. Кількісна інтерпретація результатів класифікації стосовно можливості прогнозування оцінок параметрів, аналізу та планування досліджуваних процесів.

Процедура побудови класифікації ЧР (ПКР) шляхом їх перетворення, узагальнення і вирівнювання розташованих поруч рівнів (рис. 1.5).

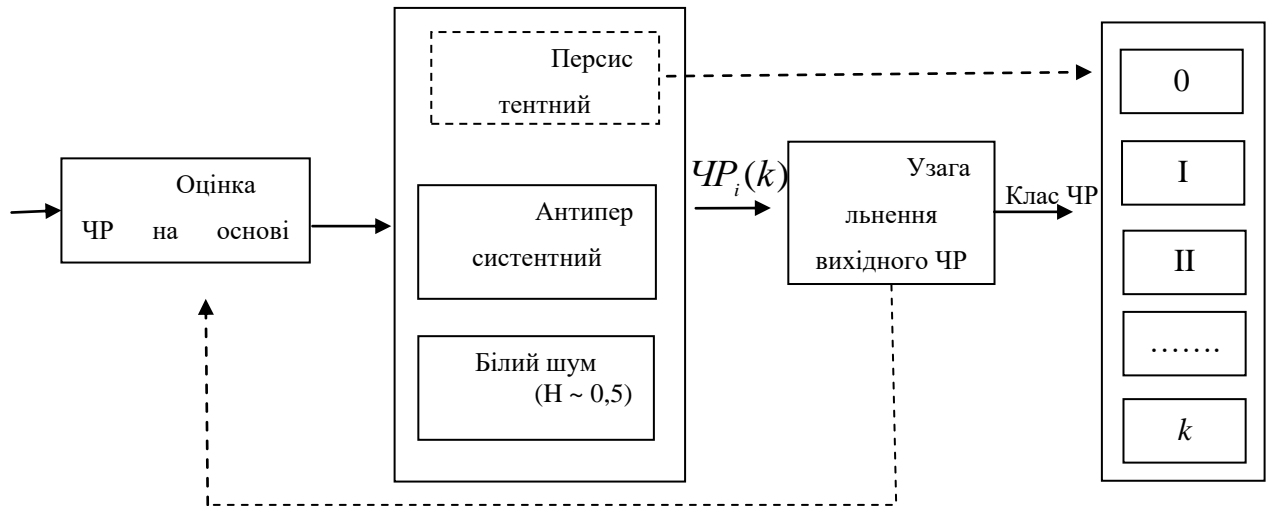


Рисунок 1.5. Схема оцінки та класифікації ЧР на основі процедури узагальнення рівнів часових рядів (ПУРАЧ)

Суть процедури класифікації (рис. 1.5) полягає в визначенні значення показника Херста для обраного ЧР. Якщо значення показника Херста ($H > 0,5$) – ЧР відноситься до класу 0. В іншому випадку виконується ПУРАЧ.

На основі вихідного ЧР формується серія $ЧР_i(k)$ нових, $k = 2, 3, 4, 5 \dots$. У серії $ЧР_i(k)$ параметр k вказує кількість послідовно розташованих рівнів ряду, які використовуються для побудови одного чергового рівня перетвореного ряду (як середнього значення рівнів k) на етапі аналізу i . Тобто « i » означає номер етапу процедури ПКР по утворенню та дослідженню перетворених $ЧР_i(k)$ на основі моделі (1.1). На наступному ($i + 1$) і подальших етапах процедура ПКР застосовується до нових, утворених на

попередніх етапах $ЧР_i(k)$. Побудова серій $ЧР_i(k)$ припиняється, якщо для деякого значення k відповідний $ЧР_i(k)$ стане персистентним відповідно (2.1). В якості іншої, але подібної умови закінчення процедури ПКР, служить виконання вимоги на етапі i : $H(ЧР_i(k)) > H * (2.1)$. При виконанні умови зупинки серії $ЧР_i(k)$ для декількох узагальнених рівнів k , будемо вважати, що $ЧР$ відноситься до класу з меншим k . Особливості реалізації умови закінчення процедури ПКР обговорюються нижче.

До отриманих на етапі k модифікованим рядах далі застосовуємо методи кореляційно-регресійного аналізу будуюмо регресійну модель утвореного $ЧР$. Мірою вірогідності кожної з побудованої є коефіцієнт детермінації (R^2). Чим він вищий, тим більше достовірна обрана модель регресії.

В роботі було удосконалено науково-методичний підхід до класифікації ТЕП, представлених АЧР спостережень, що відрізняється процедурою класифікації упорядкованих даних та дозволяє виявити додаткові ознаки зазначених процесів, за рахунок чого підвищити достовірність і точність результатів прогнозування. Практичне значення одержаних при цьому результатів полягає в тому, що запропоновані в них підходи та практичні рекомендації можуть бути використані для підвищення технологічної та економічної ефективності автоматизованого управління НДП ЗТУ за рахунок удосконалення математичних моделей і методів аналізу НДП щодо підвищення якості інформаційного забезпечення систем керування, а також удосконалити процедури прогнозування.

1.1.3. Огляд програмних аналогів

Досліджуючи весь спектр програм аналізу недетермінованих часових послідовностей (НЧП) не було знайдено загально відомих засобів моделювання та аналізу антиперсистентних часових рядів (АЧР), як окремого об'єкту досліджень. Це можна пояснити тим, що завдання утворення певної класифікації АЧР являється новим, а його моделі і методи та спеціалізовані

програмні засоби ще не уведені до найбільше відомих класичних програмних пакетів. Разом з тим існує досить багато програмних середовищ, які можливо використовувати при дослідженні АЧР. Відомі програмні засоби умовно можна розділити на 4 основні групи – розширення для електронних таблиць Excel; - додаткові модулі для Excel або окремі додатки на базі нейронних мереж; - системи статистичного аналізу, які мають вбудовані або підтримують додаткові модулі для аналізу часових рядів, бізнес-моделювання та прогнозування; - автономні додатки для моделювання і прогнозування бізнес-процесів, економетричного та інших видів аналізу даних, а також для вирішення завдань оптимізації і прийняття рішень.

Одним із сучасних підходів до моделювання недетермінованих часових послідовностей також являються моделі нечітких часових послідовностей FTS. Як АЧР, так і FTS об'єкти, досить нові, тому прямих програмних аналогів, які задовольняють вимогам, майже немає. Для аналізу придатності було обраним програмне забезпечення Oracle Crystal Ball. Oracle Crystal Ball являється додатком до Excel для інтелектуального моделювання бізнес-процесів, прогнозування невизначених змінних тощо. Використовується як додаток до Microsoft Office Excel. На рис. 1.6 зображене меню цього продукту

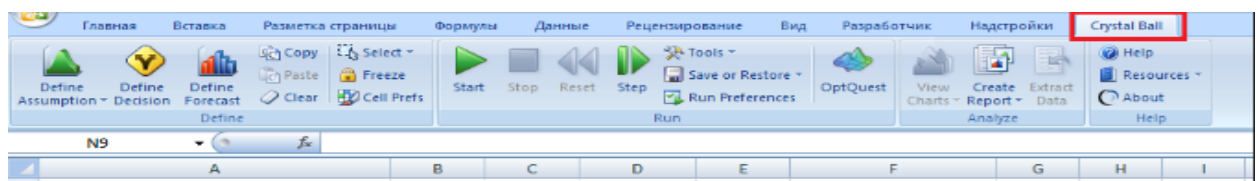


Рисунок 1.6 – Структура головного меню Crystal Ball

В засобах Crystal Ball для моделювання та прогнозування задають історичний ряд даних, внашому випадку це АЧР, а також період прогнозування, кількість наступних рівнів, рис. 1.7.



Рисунок 1.7 – Моделювання та прогнозування часових рядів

Приклад налаштування параметрів процедури прогнозування зображено на рис. 1.8. При цьому відбувається вибір діапазону історичних даних, які завантажені в систему для розрахунків прогнозу.

The screenshot shows a dialog box titled 'Set Up Form - 9999.01 Факт прошлых лет'. It has a left sidebar with 'Data Source' selected. The main area is titled 'Select source and date range of historical data'. Under 'Historical data source', 'Plan Type' is set to 'D99SPin (default)'. Under 'Date range', the 'Selected date range' radio button is selected. The 'Start year' is 'FY12', 'End year' is 'FY14', 'Start period' is 'M01', and 'End period' is 'M12'. Buttons for 'Help', 'Reset', 'Set Default', 'OK', and 'Cancel' are at the bottom.

Рисунок 1.8 – Приклад реалізації завдання із налаштування прогнозування

Засоби моделювання НЧП що розглядаються дозволяють оцінювати одну із важливих складових завдвнь із прогнозування ЧР, а саме випадкову частку помилки в розрахованих моделях даних. Вона не пояснюється формулою моделі прогнозу або тенденцією процесу. В системі Crystal Ball така похибка вимірюється пристосуванням балів за часові періоди до вхідних «історичних» даних. Після таких розрахунків вини порівнюються для величин пристосованих точок до «історичних». Можливо визначати різні форми відхилень при оцінювання рівня помилок. Тому у програмі можливо

обирати тип помилок. Наприклад, середньоквадратична помилка RMSE, являє абсолютну міру помилки прогнозу, яка ураховує позитивні та негативні відхилення процесу від трендової моделі. Іншим показником є середнє абсолютне відхилення, MAD, яка також визначає абсолютну міру помилок. Цей показник є найбільш точним для нормально розподілених даних, а також є досить надійний і широко використовується, Середня абсолютна помилка відсотка MAPE являє приклад відносної міри помилки при використанні абсолютних значень рівнів ЧР. Особливість цього показника в тому, що він дозволяє порівняти точність прогнозу між різними масштабами даних ЧР, тому що відносні помилки не залежать від масштабу залежної змінної.

В програмній системі передбачено встановлення інтервалів прогнозування: 10% та 90%; 5% та 95%; 2,5% та 97,5%; 0,5% та 99,5%; а також спеціальний: 20% та 80%. Після вибору параметрів та налаштування отримують модель у вигляді графіка, рис. 1.9.

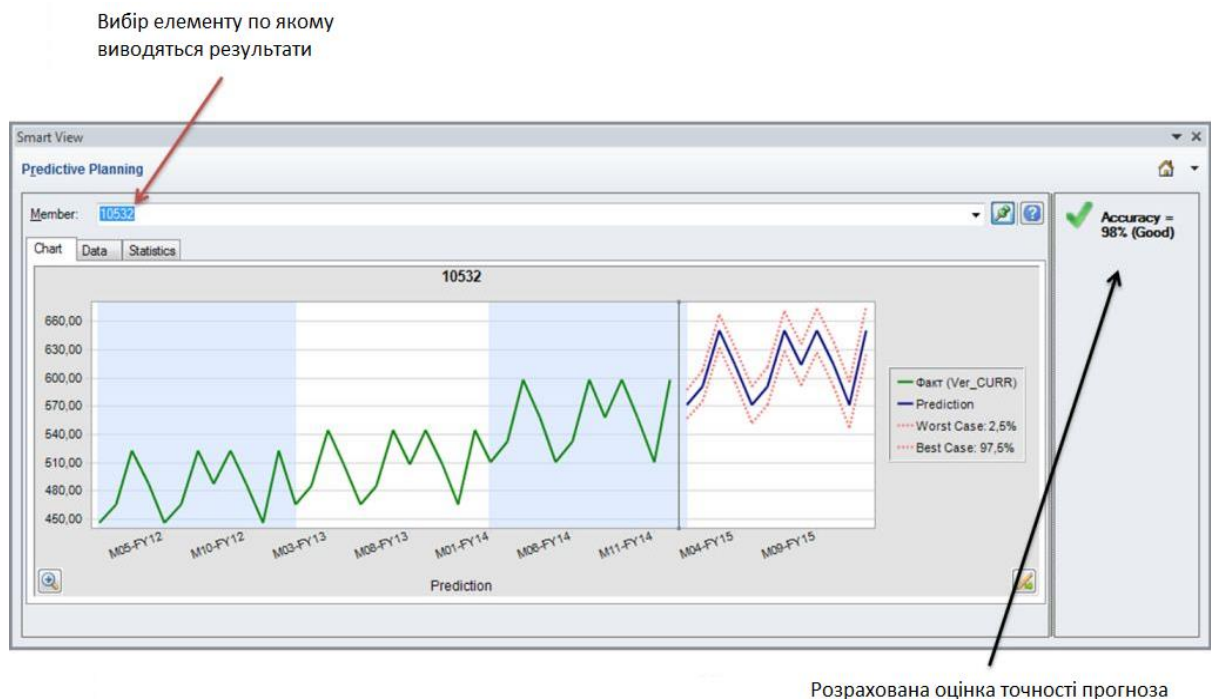


Рисунок 1.9 – Представлення графіка прогнозу часового ряду в Crystal Ball

Аналіз програмних аналогів показав, що продукт Oracle Crystal Ball досить універсальний та простий у використанні, тому що є додатком для

системи Excel. Для його застосування в завданнях аналізу та іпрьогнозування НПД достатньо мати базові навички із використання Excel. Мистема проста та гнучка в налаштуванні, має декілька типів процедур прогнозування, але разом з тим являється платною. Все ж головним являється відсутність в системі Crystal Ball спеціалізованих засобів для коректного моделювання, прогнозування та аналізу антиперсистентних ЧР, які розглядаються в нашій роботі.

1.2.Призначення та сфера застосування

На основі проведення досліджень та виконання порівняльного аналізу щодо чисельної ефективності алгоритмів класифікації та короткострокового прогнозування антиперсистентних часових послідовностей спостережень необхідно отримати рекомендації щодо можливості удосконалення процедури аналізу зазначених процесів, а також розробити чисельно ефективні алгоритми і інструментальні програмні засоби із реалізації завдань класифікації, аналізу та короткострокового прогнозування за моделями процесів АЧР.

1.3.Постановка задачі

З метою підвищення ефективності та точності методів та алгоритмів класифікації, моделювання та прогнозування антиперсистентних часових послідовностей даних спостережень необхідно провести дослідження схем агрегування часових послідовностей даних процесів, які забезпечують побудову коректних трендостійких моделей процесів, а також розробити комп'ютерні процедури для інтерполяції рівнів прогнозних показників, отриманих за агрегованими моделями.

При цьому необхідно розробити алгоритмічні та програмні засоби та виконати порівняльний аналіз чисельної ефективності алгоритмів класифікації антиперсистентних часових послідовностей спостережень, розробити алгоритмічні та програмні засоби та виконати короткострокове прогнозування антиперсистентних часових послідовностей, розробити

інструментальне програмне середовище, яке забезпечує коректне дослідження алгоритмів моделювання антиперсистентних часових послідовностей; виконати порівняльний аналіз алгоритмів моделювання антиперсистентних часових рядів щодо точності та чисельної ефективності; отримати рекомендації стосовно процедур моделювання антиперсистентних часових послідовностей.

Практичне значення мають результати проведеного аналізу стану застосування методів та засобів комп'ютерного моделювання антиперсистентних часових послідовностей спостережень, розробки та порівняльного аналізу алгоритмів моделювання, результати порівняльного аналізу чисельної ефективності алгоритмів класифікації антиперсистентних часових послідовностей спостережень, рекомендації стосовно процедур моделювання антиперсистентних часових послідовностей, а також аналізу антиперсистентних процесів. Дослідження завдань щодо порівняльного аналізу алгоритмів класифікації та моделювання антиперсистентних часових послідовностей автору не відомі.

Висновки до розділу 1

Аналіз технологічних, економічних та інших процесів, представлених недетермінованими часовими рядами спостережень, показав що існує досить широких клас ЧР, які є антиперсистентними. Для цих процесів оцінювання алгоритмів аналізу та прогнозування показників досліджені не достатньо. Натепер запропоновані методи щодо класифікації антиперсистентних процесів потребують подальшого аналізу та удосконалення, з метою забезпечення їх коректності та чисельної ефективності. Серед завдань моделювання АЧР важливим та актуальним є завдання із розробки процедури та програмного інструментарію, призначеного для оцінювання чисельної ефективності та забезпечення функції інтерполяції рівнів прогнозних показників, отриманих за агрегованими моделями.

2. ОБГРУНТУВАННЯ ТА МЕТОДИКА ДОСЛІДЖЕННЯ АЛГОРИТМІВ ІЗ МОДЕЛЮВАННЯ ТА ПРОГНОЗУВАННЯ АНТИПЕРСИСТЕНТНИХ ПРОЦЕСІВ

2.1.. Дослідження ефективності алгоритмів із моделювання та прогнозування антиперсистентних процесів.

2.1.1.. Процедури аналізу та класифікації антиперсистентних процесів

Процедури інтерполювання для комп'ютерного аналізу моделей ЧР, як оцінювання показників рівнів процесів усередині інтервалів, встановлених при класифікації часових рядів. Наведе вище процедура класифікації ЧР призначена для підвищення достовірності щодо моделей складних процесів. Вони в цілому забезпечують чітко визначену процедуру формування моделей, які дають більш обґрунтований та достовірний прогноз за розрахунок застосування спеціально утворених ЧР шляхом визначення його класу ($k = \overline{1, m}$). Разом з цим в багатьох практичних випадках результати розрахунків, отриманні засобами агрегованого ряду, не можуть використовуватися безпосередньо для завдань оперативного прогнозування, планування тощо. Для організації діяльності та інше бажано мати не лише групові дані за певний період, а оцінки рівнів для всіх періодів аналізу. При цьому із збільшенням класу ЧР використання інтервальних прогнозів для ЧР класу $k > 2$ значно ускладнюється. Через такі умови щодо можливості використання результатів прогнозування в роботі запропоновано процедуру інтерполяції, призначену для розрахунку прогнозних рівнів показників всередині класифікованого ЧР. Для цього наведемо приклад для аналізу ЧР ($k=2$), при цьому будемо визначати складові елементів для кожного рівня нового ЧР (для $k=2$ — позначимо їх відповідно Δ_1, Δ_2). Шляхом агрегації початкових рівнів вихідного ЧР Для цього побудуємо похідні ЧР(k_2),

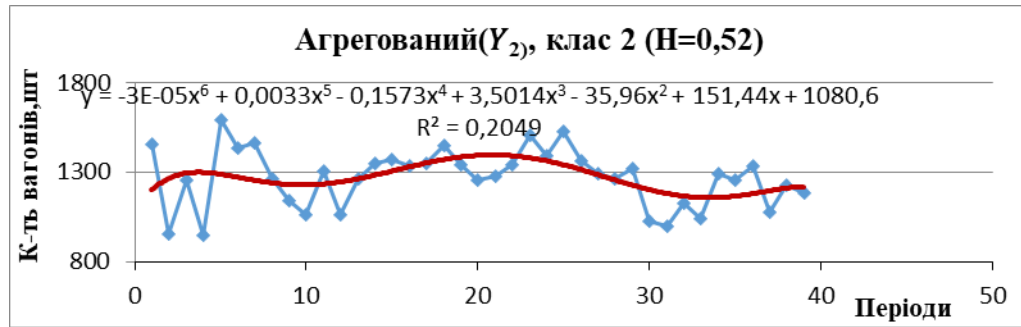
починаючи з 2-го елементу послідовності. При цьому нові утворювані $ЧР_1(k_2)(Y_1)$ та $ЧР_2(k_2)(Y_2)$ будуть формуватися із елементів, які зміщені один між собою на один на 1 рівень, певним чином «перекриваються». Тоді кожен рівень початкового ЧР ураховується при формуванні двічі, буде включений в кожен з двох похідних $ЧР_1(k_2)(Y_1)$ та $ЧР_2(k_2)(Y_2)$. Елементи початку нових ЧР розраховуються за співвідношеннями (2.1), з використанням оцінок Δ_1 та Δ_2

$$Y_1(1) = \Delta_1 + \Delta_2 \quad Y_2(1) = \Delta_2 + \Delta_3 \quad (2.1)$$

Через те що елемент Δ_1 є складовим $ЧР_1(k_2)$, можливо розрахувати значення Δ_2 . Така сама процедура застосовується для наступного етапу. Тут величина параметру Δ_2 використовується для визначення величини Δ_3 і тд. Таким чином поетапно визначаючи кожен наступний рівень на основі попередніх значень можливо виконати інтерполяцію, розрахувати всю послідовність рівнів у певному інтервалі показників, загальна середня оцінка яких визначена при агрегуванні часових рядів $ЧР_1(k_2)(Y_1)$ та $ЧР_2(k_2)(Y_2)$..

В роботі було показано результативність зазначеної процедури інтерполяції у інтервалах агрегування (ПІА), основаної на використанні зміщених послідовностей. Її результати продемонстровано рис. 2.1 та рис. 2.2. На рисунку агреговані ряди мали однаковий клас, але різні характеристики (R^2) щодо достовірності (адекватності) утворених моделей трендів.



Рисунок 2.1. Прогноз для $ЧР_1(k_2) (Y_1)$ Рисунок 2.2. Прогноз для $ЧР_1(k_2) (Y_2)$

На рис. 2.1 – рис. 2.2 для побудовано прогноз отриманого з використання зміщених послідовностей ЧР, який порівняний з прогнозом початкового АЧР. В них агреговані ряди процесу перевезень мають один клас, але різні характеристики достовірності утворених моделей трендів (R^2). При цьому для вихідного ряду $H=0,423$ ($R^2=0.067$).

Слід відзначити, що модель процесу побудована на основі агрегованого ЧР мала суттєві переваги, про що свідчать показники точності та достовірності (R^2). Такі властивості забезпечують покращення оцінок точності та достовірності отриманих прогнозних показників досліджуваних процесів, що визначається шляхом співставлення оцінок прогнозу з результатами застосування моделі прогнозу для початкового ЧР.

Для того щоб застосовувати зазначену процедуру ПІА до оцінювання та уточнення внутрішніх рівні ЧР з визначеним класом на основі (2.1) для ЧР з більш високими класами $k > 2$, необхідно використати наведену нижче систему залежностей, утворених на основі початкового ЧР:

$$\begin{cases} Y_1(1) = \left(\frac{1}{k}\right) \sum_{i=1}^k \Delta_i \\ Y_2(1) = \left(\frac{1}{k}\right) \sum_{i=2}^{k+1} \Delta_i \\ \dots \\ Y_k(1) = \left(\frac{1}{k}\right) \sum_{i=k}^{k+k} \Delta_i \end{cases} \quad (2.2)$$

Відповідно (2.2) для розрахунку оцінок рівнів показника всередині інтервалу агрегування для недетермінованих процесів, представлених числовими послідовностями ЧР класу $k > 1$, необхідно утворити « k » нових послідовностей, елементи яких розраховуються шляхом агрегування та визначення середніх значень послідовностей рівнів у рамках діапазонів індексів. Наступні елементи послідовностей агрегованих моделей розраховують подібно (2.2), але для інших послідовностей індексів елементів ЧР. Наприклад для $YI(2)$ $i = (k+1)$ до $i = 2k$. При цьому система індексів номерів в різних агрегованих моделях зміщена відносно інших. Утворена за рахунок наборів процедура прогнозування внутрішніх рівні ЧР ПІА дозволяє «уточнити» всі значення. За рахунок чого виникає можливість щоби надати додаткову інформацію для аналізу та планування недетермінованих процесів. Остаточно визначається що процедура ПІА шляхом формування і використовуючи « k » агрегованих трендів, які «зміщені» за індексами (агрегованих) ЧР, $ЧР_i^{(k)}(Y_{2i})$ дозволяє отримати прогностичні значення для інтервальних рівнів ($k = \overline{2, P}$), причому в середині самого інтервалу визначеного класу ЧР, тобто виконувати функцію інтерполяції.

В роботах також запропоновано процедури щодо порівняльного аналізу розрахованих уточнених рівнів ЧР в середині інтервалів. При цьому співставляють значення отримані за процедурою ПІА та значення «посередньому», коли вважаються всі внутрішні рівні однаковими. Було встановлено що середня абсолютна похибка прогнозу у відсотках ($MAPE$) приблизно однакова для ЧР класу « $k=2$ » та для прогнозу по «посередньому», Методи та процедури інтерполяції щодо уточнення рівнів ЧР на основі ПІА визначеного класу в цілому мали переваги, в порівнянні з точністю прогнозу для початкового ЧР.

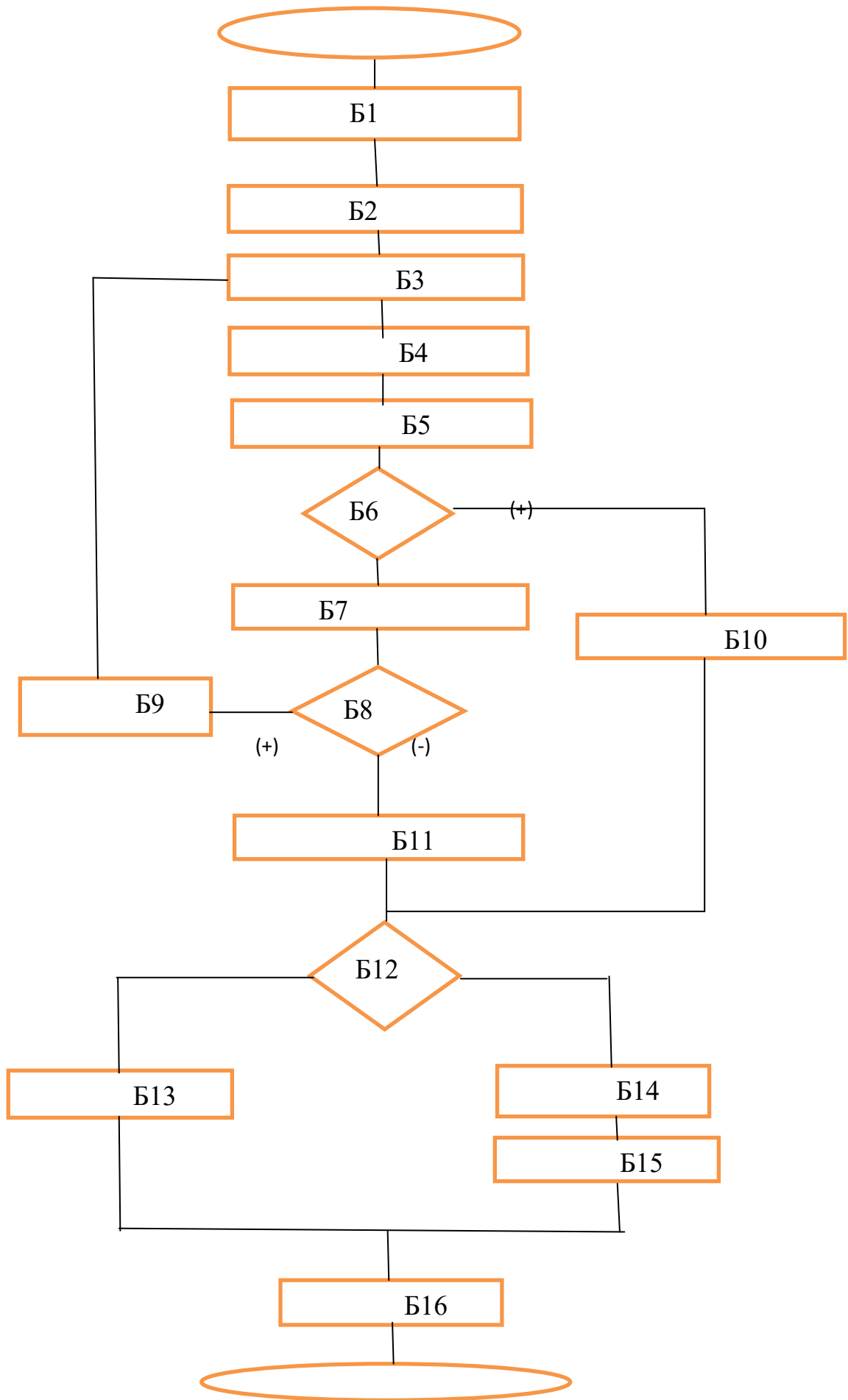


Рисунок 2.3. Блок-схема процедури дослідження класів АЧР

Блок-схема рис. 2.3 визначає загальну процедуру щодо класифікації АЧР. При цьому зміст окремих складових моделей полягає у наступному: Б1 – отримання вихідних даних для розрахунків, числа елементів ЧР, типу варіанту агрегування (з перекриттям рівнів/без перекриття), граничне значення константи Херста, що визначає кінець процесу класифікації, граничний клас агрегування, який припустимий при моделюванні; Б2 – визначення початкових параметрів стосовно номеру класу та мінімального числа елементів послідовності ЧП, необхідних для моделювання; Б3 – формування агрегованого ряду для класу АЧР, який перевіряється; Б4 – процедура розрахунку параметру Херста для контрольованого агрегованого ряду; Б5 – запис величини параметру Херста у таблицю процесу для подальшого аналізу; Б6 – перевірка умови закінчення формування моделей класів, встановлення класу АЧР; Б7 – при необхідності продовження дослідження типу АЧР (визначення його класу) виконується розрахунок числа можливих членів послідовності на наступному етапі; Б8 – перевірка умови щодо припустимого числа елементів у наступній агрегованій моделі; Б9 – у разі (+) і виконання умови за кількістю членів нового агрегованого ряду змінюють на (+1) номер класу АЧР, який треба перевірити; Б10 – клас АЧР визначено, встановити прапорець «Клас=(+)» та перейти до запису результатів; Б11 - клас АЧР не визначено, встановити прапорець «Клас=(-)»; Б12 - перевірка варіанту закінчення процесу класифікації, параметру «Клас=(+/-)»; Б13 – повідомлення про відсутність класифікації АЧР, сформувати таблицю величин параметрів Херста для різних розрахованих варіантів агрегування даних АЧР; Б14 – видача номеру класу для АЧР, таблиці величин агрегованих елементів, моделі тренду для цього класу; Б15 – видача графічного представлення вихідного АЧР, а також моделі тренду для встановленого класу ряду даних.

2.1.2.. Аналіз напрямків удосконалення алгоритмів класифікації та короткострокового прогнозування антиперсистентних процесів

Результати аналізу спеціалізованих процедур аналізу та моделювання недетермінованих послідовностей величин як часових рядів (ЧР), дозволяють встановити наступне.

1) Дослідження антиперсистентних числових послідовностей (АЧР) стосовно процесів узагальнення їх рівнів, показали неоднорідність таких АЧР. Показана можливість класифікації таких АЧР. Визначені при цьому класи АЧР « k » дають підставу для формування моделей трендів та інтерпретації шляхом інтерполяції узагальнених рівнів. Можливість диференціювати антиперсистентні ЧР – віднести до різних класів (що розрізняються числом об'єднаних поруч стоячих рівнів, довжинами інтервалів достовірного планування). При цьому також можливо виконувати процедури прогнозування значень сумарних об'ємних показників модифікованого ряду за встановлений інтервал (період узагальнення даних).

Наприклад, аналіз за процедурою агрегування АЧР дозволив встановити, що представлений ЧР відноситься до 5-го класу (рис. 2.4). Тобто достовірно можливо прогнозувати (оцінювати можливі значення показника) такий процес на 5 інтервалів.

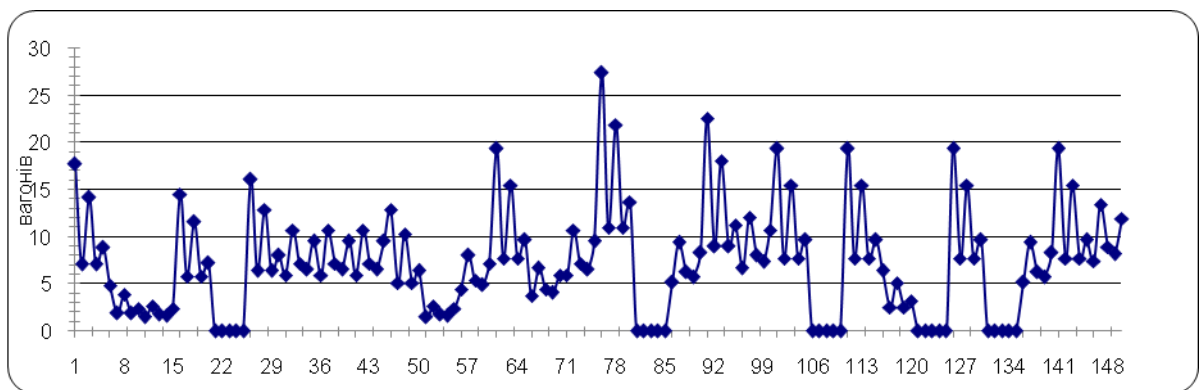


Рисунок 2.4. ЧР₁(5), $H=0,587$

2) За результатами аналізу згідно процедур агрегування властивостей модифікованих ЧР відзначено, що при збільшенні інтервалу узагальнення « k » вихідний антиперсистентний ЧР систематично переводиться в персистентний, але за різну кількість кроків. Це свідчить про збереження «власних властивостей» досліджуваного ЧР (наявність механізмів породження недетермінованих процесів в силу власних механізмів, які проявляються при певних умовах).

3) У названих роботах були сформульовані твердження (або принцип) невизначеності хаотичного ряду. За ним встановлюється можливість достовірного аналізу та прогнозування значень показників лише для певного інтервалу k , значення якого обумовлена властивостями АЧР. В рамках процедури узагальнення відкривається можливість кількісного аналізу антиперсистентних ЧР, але лише на виявленому при класифікації інтервалі. Якщо в результаті застосування процедур агрегування ЧР можна віднести до декількох класів одночасно, ЧР слід віднести до найменшого за номером « k » класу.

4) Наведені результати, подані кроки узагальнення та класифікації використовуючи конкретні ЧР, отримані у сфері реалізації вантажних залізничних перевезень, мають практичне застосування та були покладені у процедури із планування відповідних чинників утворення вагонопотоків на станціях.

5) Було встановлено, що процедури моделювання недетермінованих процесів на основі узагальнення рівнів можуть бути застосовані не лише для АЧР, а і для випадкових (нестационарних) ЧР за Херстом. Це наче розширює сферу її практичного використання в завданнях комп'ютерного моделювання у різних сферах.

6) Відзначимо, що в публікаціях виконано та приведено аналіз лише моделі агрегування виду (2.4). Разом з тим можливо запропонувати інші способи утворення агрегованих трендів, які «зміщені» за індексами. В дипломі проведені дослідження кількох способів утворення моделей шляхом

агрегування, створені алгоритми та програмні засоби для їх порівняльного аналізу, отримані рекомендації щодо чисельної ефективності. Також недостатньо досліджені завдання щодо реалізації завдань прогнозування показників ЧР на основі агрегованих часових рядів, утворених при класифікації ЧР.

2.1.3.. Методика та процедури дослідження чисельної ефективності та точності алгоритмів класифікації та інтерполяції даних антиперсистентних процесів. Порівняльний аналіз.

Відповідно до наведених вище результатів аналізу відомих процедур дослідження АЧР шляхом їх узагальнення (агрегування послідовностей рівнів), можливо отримати різні класи АЧР, в залежності від того, при якому числі рівнів агрегування « k » вперше (найменше значення « k ») виконується умова трендостійкості моделі нового ЧР – $H_{\text{ЧР}}(Y_k) \Rightarrow 0.5$. Разом з тим залишилися не дослідженими питання щодо варіантів утворення схем агрегування АЧР. Із зростанням складності АЧР, тобто номеру класу « k », зростає число можливих структур утворення агрегованих моделей процесів. На рис. 2.5 – рис. 2.6 наведено варіанти схем, за якими можливо утворювати із початкового АЧР нові агреговані. На рис. 2.5 указано, що агреговані елементи можуть бути утворені «з перекриттям» f_b , або «без перекриття» рівнів f_c початкового АЧР f_a . При цьому для моделей (f_b, f_c) можуть бути математично коректно побудовані функції, які являють тренди нових узагальнених процесів.

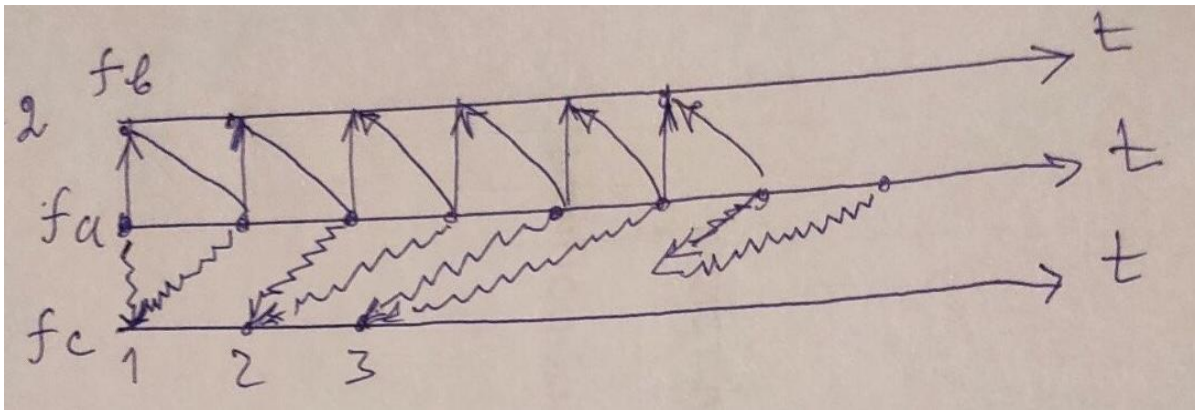


Рисунок 2.5. Схеми варіантів агрегування вихідних даних для класифікації ЧР ($k=2$),

Рис. 2.6 демонструє зростання можливих варіантів схем утворення агрегованих моделей початкового АЧР при пошуку класифікації при ($k=3$). Так для моделі f_b , початкові члени агрегованої послідовності визначаються через наступні елементи $u_{b1} = (u_{a1} + u_{a2} + u_{a3})/3$; $u_{b2} = (u_{a2} + u_{a3} + u_{a4})/3$; $u_{b3} = (u_{a3} + u_{a4} + u_{a5})/3$;

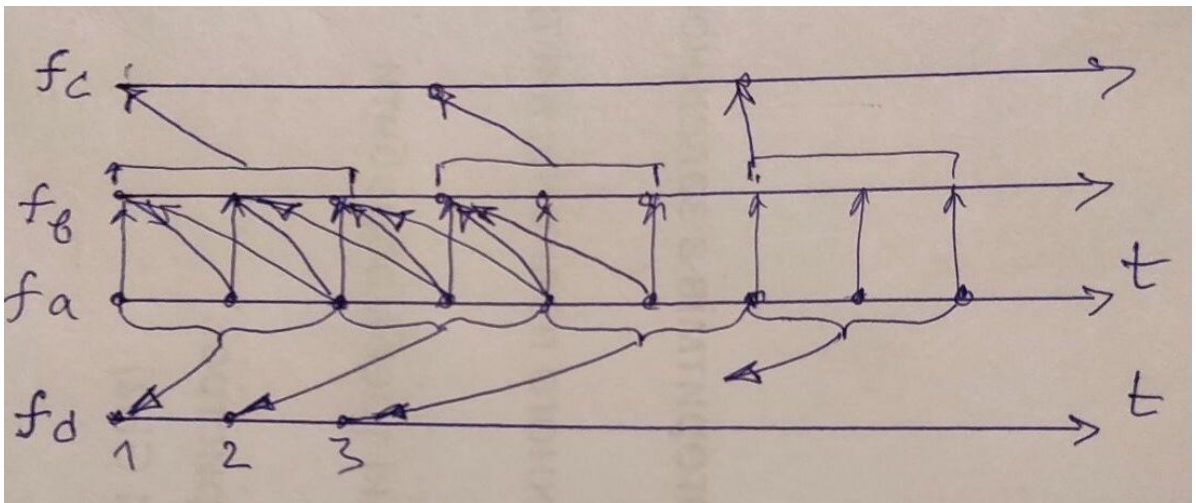


Рисунок 2.6. Схеми варіантів агрегування вихідних даних для класифікації ЧР ($k=3$),

Для моделі «без перекриття» членів АЧР f_c початкові члени агрегованої послідовності визначаються через наступні елементи

$$u_{c1} = (u_{a1} + u_{a2} + u_{a3})/3; u_{c2} = (u_{a4} + u_{a5} + u_{a6})/3; u_{c3} = (u_{a7} + u_{a8} + u_{a9})/3;$$

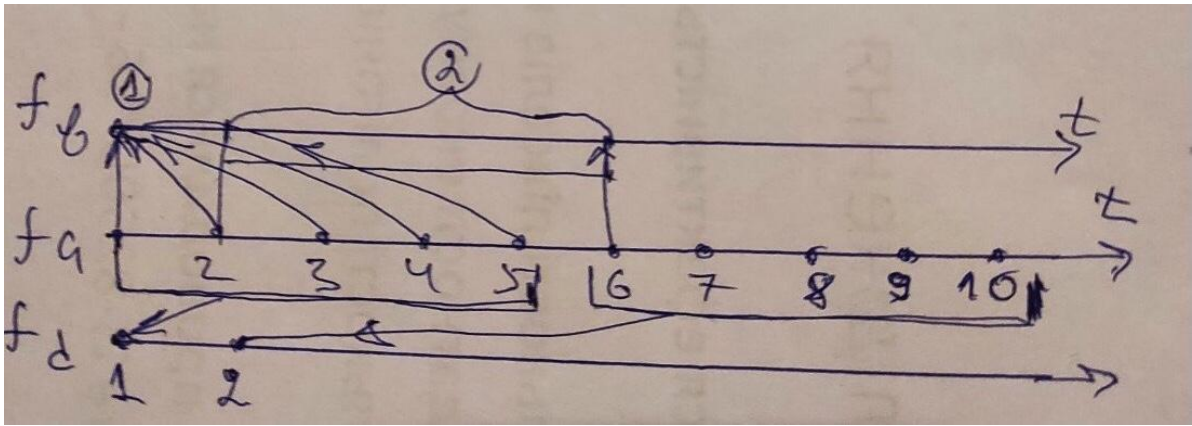


Рисунок 2.7. Схеми граничних варіантів агрегування вихідних даних для класифікації ЧР ($k=5$),

На рис. 2.7 схематично зображені граничні варіанти можливих схем утворення агрегованих послідовностей даних для визначення 5-го класу антиперсистентного ЧР. Це f_b , коли агреговані моделі містять послідовні члени, що відрізняються одним елементом

$$u_{b1} = (u_{a1} + u_{a2} + u_{a3} + u_{a4} + u_{a5})/5; u_{b2} = (u_{a2} + u_{a3} + u_{a4} + u_{a5} + u_{a6})/5;$$

$$u_{b3} = (u_{a3} + u_{a4} + u_{a5} + u_{a6} + u_{a7})/5; \dots$$

А також протилежною схемою формування нових послідовностей, коли послідовні елементи агрегованих моделей не містять попередні елементи, а саме

$$u_{d1} = (u_{a1} + u_{a2} + u_{a3} + u_{a4} + u_{a5})/5; u_{d2} = (u_{a6} + u_{a7} + u_{a8} + u_{a9} + u_{a10})/5;$$

$$u_{d3} = (u_{a11} + u_{a12} + u_{a13} + u_{a14} + u_{a15})/5; \dots$$

Наведені на рис. 2.5 – рис. 2.7 приклади варіантів агрегування даних для побудови узагальнених коректних математичних моделей недетермінованих процесів, сформованих за даними спостережень. Демонструють змістовність і широкі можливості утворення агрегованих моделей процесів у вигляді ЧР. При цьому в дослідженнях необхідно встановити, по-перше, при якій формі агрегування забезпечується можливість класифікації вихідних ЧР, а по-друге, запропонувати алгоритми прогнозування за агрегованими моделями. Такі алгоритми мають також дозволяти виконувати інтерполяцію – розраховувати внутрішні рівні показників процесів ЧР, використовуючи оцінки інтегрованих величин,

отриманих за моделями трендів виду (f_b, f_c) та подібних до них, див. рис. 2.5 – рис. 2.7.

Зазначимо також що для різних варіантів агрегування необхідно формувати окрему процедуру інтерполяції - розрахунку величин внутрішніх рівнів на основі прогнозних величин для всього діапазону, отриманих на основі системи рівнянь (2.2). Наприклад, для ($k=3$) за отриманими моделями f_c для розрахування внутрішньо інтервальних значень прогнозних рівнів можливо замість рішення системи рівнянь виду (2.2) використовувати ($k-1$) початкові значення початкового АЧР ряду Y . Тоді для розрахунку інтерполяції рівнів достатньо використовувати співвідношення типу

$$\begin{aligned} Y_1 &= \frac{y_1 + y_2 + y_3}{3}, \text{ то } \rightarrow \hat{y}_3 = 3 * Y_1 - y_1 - y_2 \\ Y_2 &= \frac{y_2 + y_3 + y_4}{3}, \text{ то } \rightarrow \hat{y}_4 = 3 * Y_2 - y_2 - \hat{y}_3 \\ Y_3 &= \frac{y_3 + y_4 + y_5}{3}, \text{ то } \rightarrow \hat{y}_5 = 3 * Y_3 - \hat{y}_3 - \hat{y}_4 \end{aligned}$$

замість рішення систем рівнянь.

Для реалізації зазначеної форми алгоритмів інтерполяції необхідно мати відповідну структуру програмної реалізації, яка забезпечує утворення модельної послідовності що містить всі попередні рівні ЧР.

2.2.. Дослідження функціональної ефективності алгоритмів моделювання.

Мета дослідження – оцінювання ефективності алгоритмів узагальнення часових рядів для забезпечення вимог щодо побудови класифікації недетермінованих процесів. Основна задача – дослідити порівняльну ефективність процедур узагальнення антиперсистентних часових рядів (АЧР) для переведення моделей отриманих процесів до персистентних класів по Херсту, тобто при виконанні умови щодо показника $N_{\text{чр}} \Rightarrow 0.5$.

В проведених дослідження показано, що різні схеми агрегування дають різні (хоча і досить схожі) математичні моделі одного ж й того ж самого процесу, які можна застосувати для оперативного прогнозування в умовах

невизначеності. Нижче наведено приклади моделей при застосуванні процедури агрегування з перекриттям рівнів.

Досліджувався ЧР представлений на (рис.8), з показником Херста, $H = 0,479$.

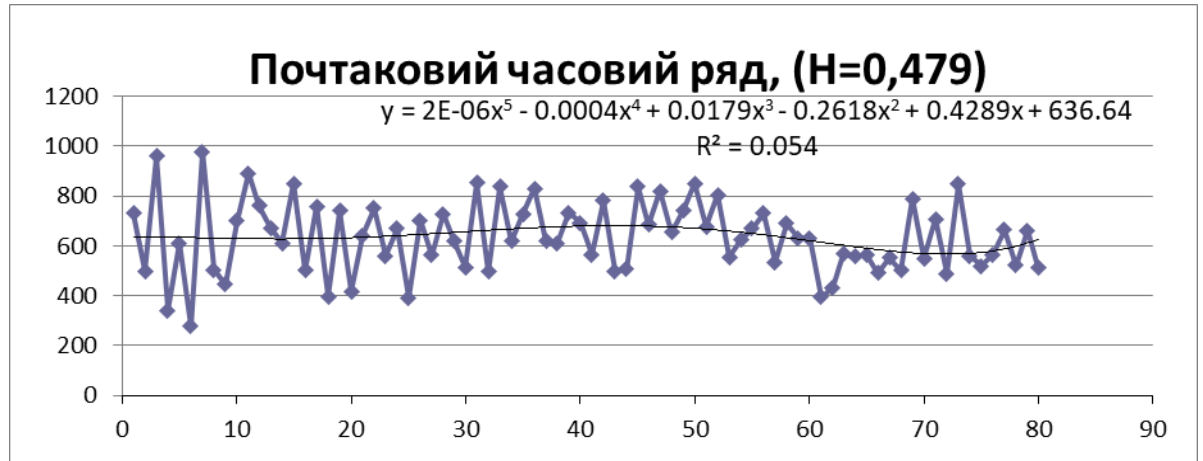
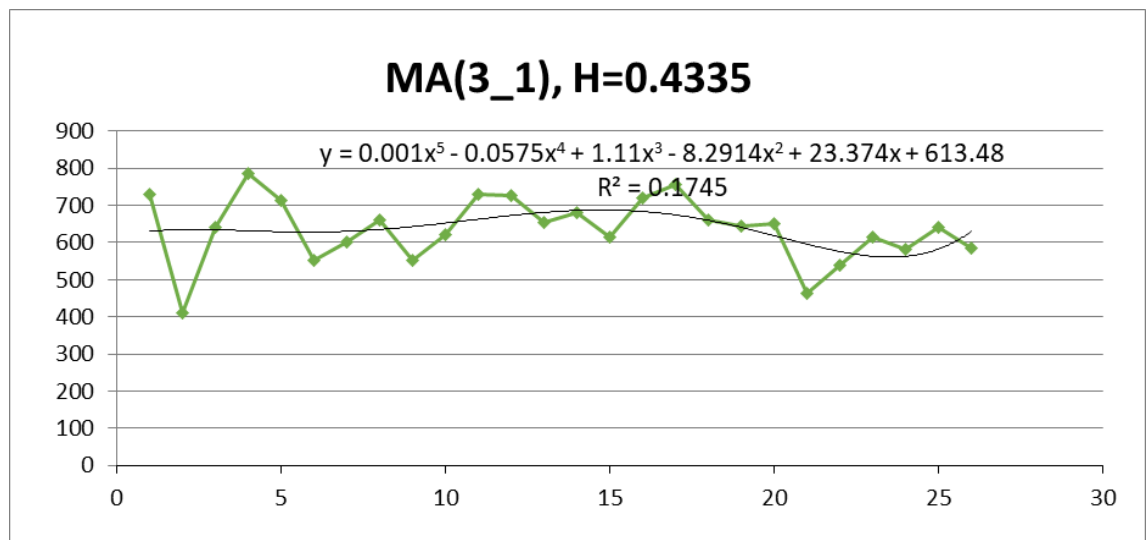


Рисунок 2.8 - Початковий АЧР

Після визначення класу ЧР [], ($k=3$) та застосування процедури агрегування [8] з перекриттям, отримані наступні моделі ЧР - рис. 2.9 – рис. 2.11



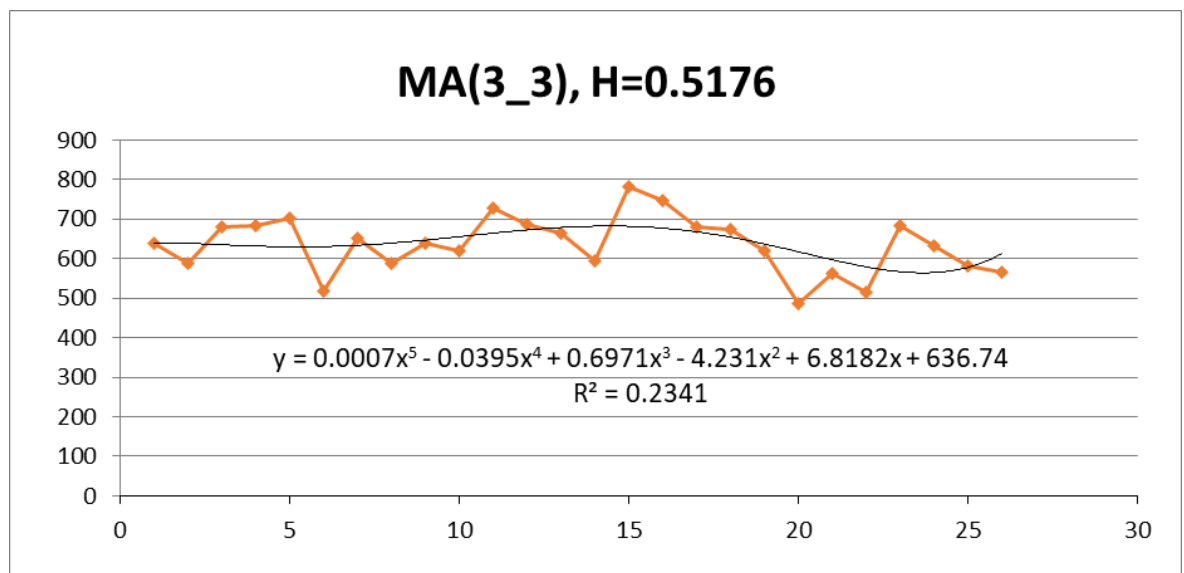
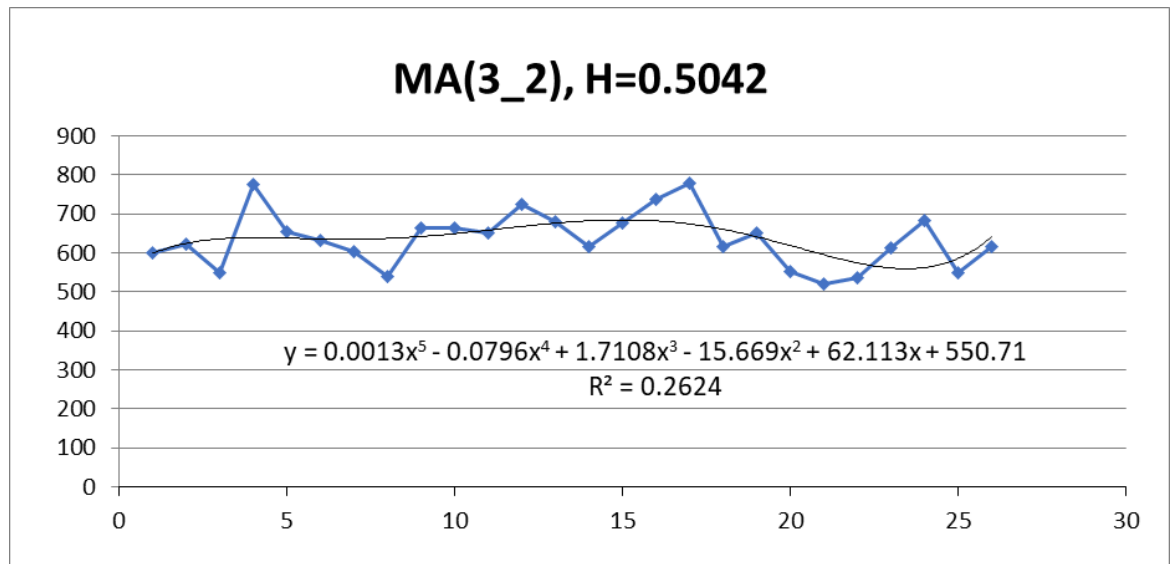


Рисунок 2.11 ЧР отримані в результаті агрегації по ($k=3$) рівнів з перекриттям з відповідними моделями та оцінкою достовірності (R^2).

Для побудови прогнозу використовується наступний алгоритм:

Для Y_1 вибирається k -рівнів початкового ЧР, починаючи з $i=1, n$ по k .

Для Y_2 – $i=2, n$ по k , і т.п., для всіх $Y_j, j=1, k$

$$Y_i = \frac{\sum_{i=1}^k Y_i}{k}$$

Визначається коефіцієнт Херста, будується прогноз та розраховується коефіцієнт детермінації (R^2).

За отриманими моделями розраховуються внутрішньо інтервальні значення прогнозних рівнів, для визначення яких використовуються (k-1) початкові значення ряду Y. Наприклад, для (k=3):

$$Y_1 = \frac{y_1 + y_2 + y_3}{3}, \text{ то } \rightarrow \hat{y}_3 = 3 * Y_1 - y_1 - y_2$$

$$Y_2 = \frac{y_2 + y_3 + y_4}{3}, \text{ то } \rightarrow \hat{y}_4 = 3 * Y_2 - y_2 - \hat{y}_3$$

$$Y_3 = \frac{y_3 + y_4 + y_5}{3}, \text{ то } \rightarrow \hat{y}_5 = 3 * Y_3 - \hat{y}_3 - \hat{y}_4$$

1. Таким чином розраховується всі наступні рівні прогнозного ряду (рис.5).

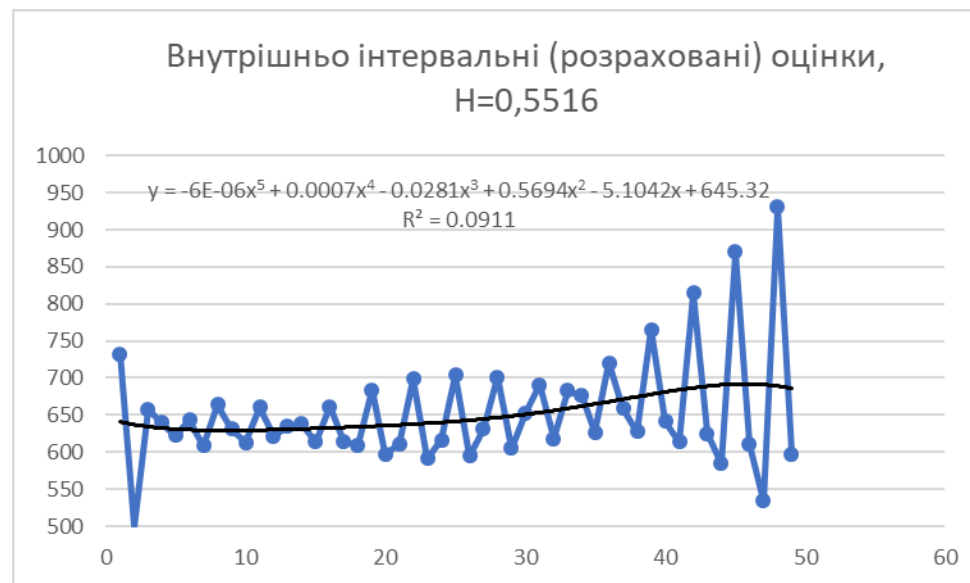


Рисунок 2.12

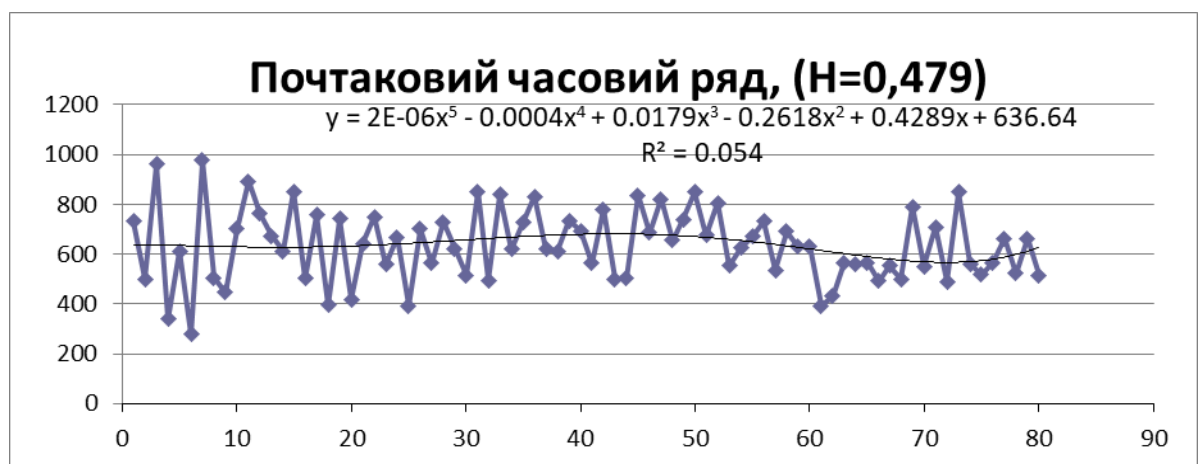


Рисунок 2.13

Наведем результати досліджень для інших варіантів схем агрегування АЧР.

23.. Особливості структури процесів моделювання антиперсистентних числових послідовностей.

Приведені результати досліджень щодо класифікації та прогнозування АЧР з використанням різних варіантів схем агрегування дозволяють розширити структури системи, подану на рис. 1.5. А саме до структури системи дослідження моделей АЧР додаються нові функції аналізу, які представлені на рис. 2.14. Перше за все відзначимо нові, відмінні від рис. 1.5 модулі системи. Це блок вибору варіантів моделей схем агрегування (М1/М2), блок формування агрегованих, узагальнених часових послідовностей відповідних типів Б4, модуль формування та дослідження математичних моделей трендів утворених узагальнених ЧР Б6. В цьому ж модулі виконуються розрахунки прогностичних показників моделей, а також інтерполяція внутрішніх рівнів на основі показників, які характеризують діапазон у цілому.

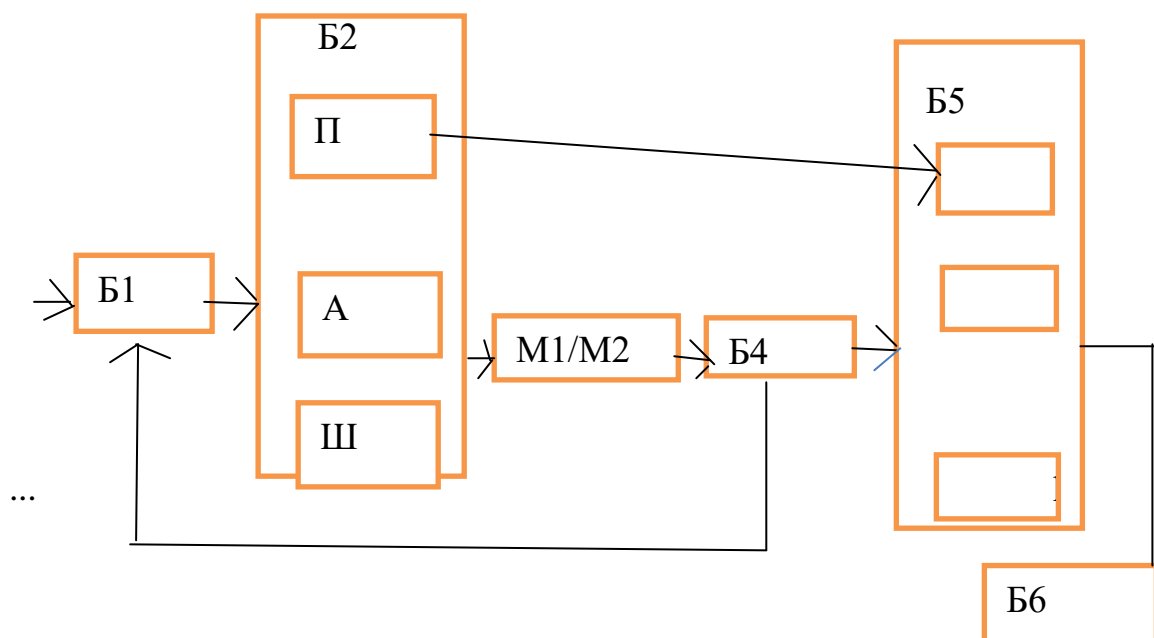


Рисунок 2.14. Структура системи дослідження на прогнозування показників АЧР.

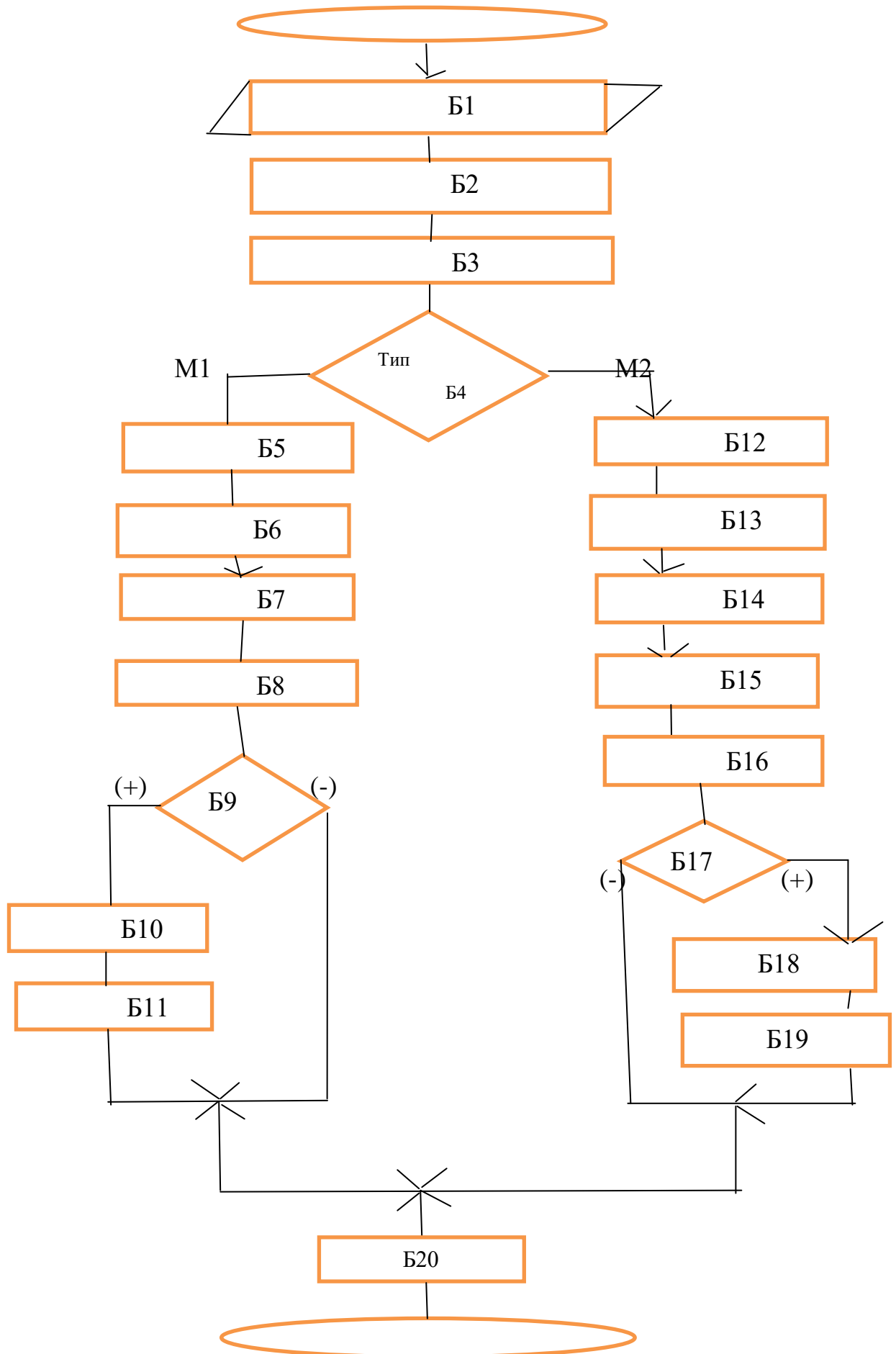


Рисунок 2.15. Схема процедури моделювання і прогнозування антиперсистентний часовий ряд

Функції інших модулів (блоків системи) наступні. В модулі Б1 виконується оцінка константи Херста для заданого ЧР. Під блоки для Б2 визначають категорію ЧР, а саме П (персистентний), А (антиперсистентний), Ш («шум» невизначений). Б5 забезпечує функцію встановлення класів « k » послідовності даних АЧР.

На рис. 2.15 наведено узагальнену блок-схему яка визначає складові процесів моделювання на основі агрегування рівнів, як наведено вище, а також прогнозування за утвореними математичними моделями трендів для узагальнених ЧР. Схема передбачає дві основні категорії моделей М1 – «з перекриттям» рівнів та М2 – «без перекриття», рис. 2.5, рис. 2.6.

Блоки схеми рис. 2.15 реалізують наступні функції: Б1 – отримання вхідного ЧР, підрахунок числа елементів; Б2 – процедура розрахунку параметра Херста; Б3 – процедура класифікації, визначення властивостей (П/А/Ш рис. 2.14) первинного ЧР; Б4 – вибір типу моделей агрегування (М1/М2) рис. 2.14; Б5 – формування послідовностей із агрегованих членів початкового ЧР, рис. 2.5 – рис. 2.7; Б6 – формування моделей для трендів сформованих агрегованих ЧР; Б7 – розрахунок коефіцієнту детермінації R^2 для моделей трендів; Б8 – розрахунок інтегрованого показника точності моделей трендів; Б9 – умова щодо потреби виконання розрахунку прогнозних величин за моделями трендів; Б10 – розрахунок прогнозних оцінок процесу, а також процедура інтерполяції щодо визначення внутрішніх рівнів процесу в області прогнозу; Б11 – формування табличного відображення прогнозних оцінок; Б20 – збереження даних; Б12 - формування визначених варіантів та числа послідовностей із агрегованих членів початкового ЧР, рис. 2.5 – рис. 2.7 формування послідовностей із агрегованих членів початкового ЧР, рис. 2.5 – рис. 2.7; Б13 - формування визначеного числа моделей для всіх трендів сформованих агрегованих ЧР; Б14 - розрахунок коефіцієнтів детермінації R^2 для всіх сформованих моделей

трендів первинного АЧР; Б15 – визначення максимального R^2 серед побудованих моделей трендів, відбір кращої моделі тренду та відповідного варіанту схеми агрегування; Б16 - розрахунок інтегрованого показника точності для кращої із моделей трендів; Б17 – умова щодо потреби виконання розрахунку прогнозних величин за кращим із трендів; Б18 - розрахунок прогнозних оцінок процесу, виконання процедури інтерполяції для внутрішніх рівнів процесу в області прогнозу; Б19 – формування табличного відображення прогнозних оцінок.

Висновок до розділу 2

У розділі проведено дослідження можливостей та ефективності алгоритмів із моделювання та прогнозування показників антиперсистентних процесів. При цьому розглянуто структури процедур аналізу та класифікації антиперсистентних процесів, проведено аналіз напрямків щодо удосконалення алгоритмів класифікації та короткострокового прогнозування антиперсистентних часових послідовностей, запропонована методика та процедури дослідження АЧР шляхом порівняльного аналізу чисельної ефективності та точності алгоритмів класифікації та інтерполяції даних антиперсистентних процесів. На основі зазначеної методики класифікації та прогнозування АЧР проведені дослідження функціональної ефективності алгоритмів моделювання. В розділі були визначені особливості структури процесів моделювання антиперсистентних числових послідовностей. При порівнянні алгоритмів класифікації АЧР було встановлено, що найбільше детальною та сталою являється класифікація на основі схем агрегування без перетину елементів, наведена у розділі. Отримані результати далі використані при формування програмного забезпечення для дослідження моделей дискретних антиперсистентних процесів.

3. РОЗРОБКА ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ДЛЯ ДОСЛІДЖЕННЯ МЕТОДУ МОДЕЛЮВАННЯ ЧАСОВОЇ ПОСЛІДОВНОСТІ

3.1.Формалізація задачі

Формалізація задачі представлена на рівні зовнішнього проектування наведена у вигляді діаграми варіантів використання. Користувач представлений у вигляді актора-дослідника, що взаємодіє із системою за допомогою варіантів використання. Варіанти використання надають опис можливостей, які система надає акторам.

Діаграма прецедентів визначає варіанти використання системи аналізу процесів моделювання ЧДП, що зображено на рис. 3.1

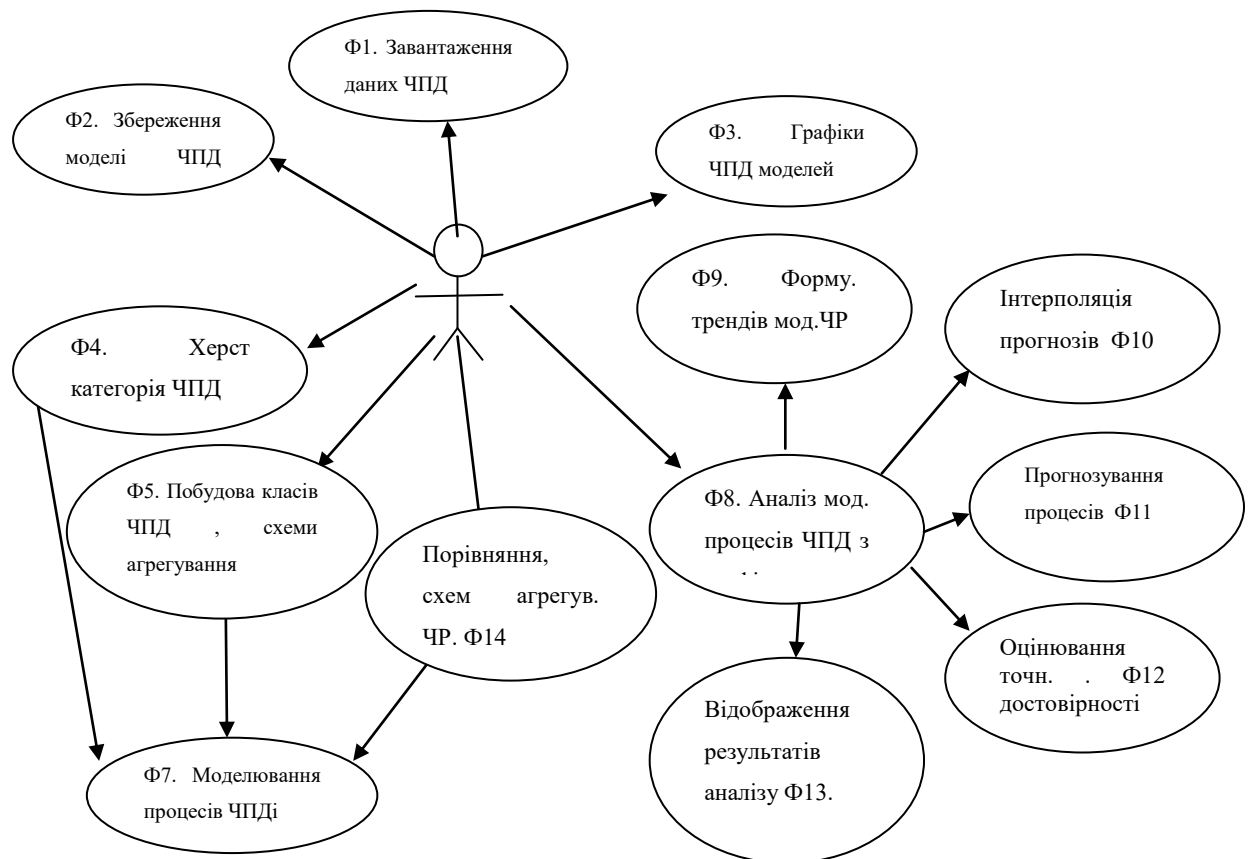


Рисунок 3.1 – Діаграма варіантів використання при дослідженні ЧР

На діаграмі представлені наступні функції програми: Ф1 – процедури отриманні вихідних даних для дослідження, як із текстового файлу, так і в форматі Excel; Ф2 – функція щодо збереження даних та результатів

досліджень; Ф3 – відображення графіків моделей процесів; Ф4 – розрахунки оцінок показників Херста для Часових Послідовностей Даних (ЧПД); Ф5 – побудова класів для моделей ЧПД, використовуючи різні схеми агрегування послідовностей даних; Ф6 – порівняльний аналіз варіантів схем агрегування ЧР; Ф7 – моделювання процесів, представлених ЧПД; Ф8 – Аналіз моделей процесів ЧПД; Ф9 – процедури із форму. трендів моделей процесів ЧПД; Ф10 – інтерполяція внутрішньо інтервальних даних для оцінок прогнозів за моделями ЧПД; Ф11 – прогнозування процесів за моделями ЧПД; Ф12 – процедури оцінювання точності та достовірності (розрахунок коеф. R^2) моделей ЧПД; Ф13 – процедури відображення результатів аналізу моделей процесів ЧПД; Ф14 – процедури порівняння, виконання аналізу схем агрегування ЧПД.

3.2.Базова архітектура програми

В якості базової архітектури програмного комплексу «Дослідження моделей антиперсистентних часових послідовностей». використана архітектура бібліотеки VCI (Visual Component Library, VCL). Бібліотека візуальних компонентів VCI це об'єктно-орієнтована бібліотека, яка призначена для розробки програмного забезпечення. Вона розробляється компанією Borland (підтримується Embarcadero) для підтримки технологій візуального програмного забезпечення. Бібліотека VCL входить до комплексу Delphi, C++ Builder та Embarcadero RAD Studio, а також VCL знаходиться в комплекті з окремими розробниками ПЗ, які реалізують запропоновані в цих програмних середовищах місця без використання VCL. Бібліотека VCL надає значну кількість готових до використаних компонентів, пристосованих для роботи в самих різних обласних програмах. Наприклад, при створенні інтерфейсів користувача (екранні форми і елементи управління - т. Н. «Контролі»), при роботах з використанням баз даних, при організації взаємодії з операційними системами, для програмування мережевих застосувань ПЗ та інших.

Взаємодія головних елементів між частинами архітектурної моделі VCL у рамках системи «Дослідження моделей антиперсистентних часових послідовностей», представлено на рис. 3.2

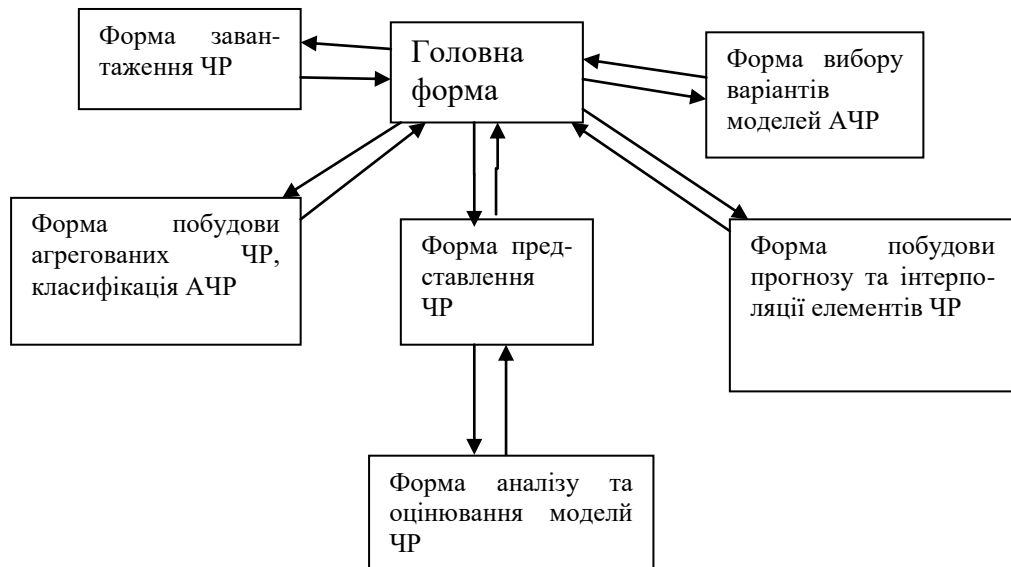


Рисунок 3.2 – Взаємодія головних елементів архітектурної моделі VCL програми «Дослідження моделей антиперсистентних часових послідовностей».

3.3. Внутрішнє проектування

3.3.1. Вибір мови програмування

При виборі мови програмування урахувалися фактори стосовно можливості програмного середовища, а також можливостей безпосередньо мови програмування. Саме ці два головних критерії урахувалися при визначенні мови програмування.

Для розробки програмного комплексу було обране програмне середовище для зручної розробки, як функціоналу так і інтерфейсу користувача. При цьому було обрано середовище RAD Studio Builder. Ця система надає великі можливості для швидкої розробки інтерфейсу користувача, має велику базу готових компонентів, які узгоджені між собою, забезпечують ефективну їх взаємодію з одним з одним. Також

середовище RAD Studio Builder забезпечує зручне налагодження програми, засоби для знаходження логічних та синтаксичних помилок тощо.

При розробках програми «Дослідження моделей антиперсистентних часових послідовностей» використано мову програмування C++, тому що компоненти середовища RAD Studio Builder також написані на C++. До уваги прийнято і те, що C++ підтримує різноманітні парадигми програмування, а саме процедурне програмування, об'єктно-орієнтоване програмування, узагальнене програмування. Засоби мови мають надзвичайно багату стандартну бібліотеку, яка містить поширені контейнери і алгоритми, процедури введення-виведення, апарат регулярних виразів, підтримку багато поточності і інші можливості. Важливою особливістю мови C++ у тому, що вона поєднує властивості як високорівневих, так і низькорівневих мов програмування. На відмінність від свого попередника, мови C, у C++ найбільшу увагу приділено засобам підтримки об'єктно-орієнтованого і узагальненого програмування.

3.3.2. Ієрархія та взаємодія класів програмної системи

Взаємодія та проектування класів описано UML діаграмою.

Клас Main являється головним вікном програмного комплексу та породжує вікна ftschart, ftsms, editftschart, exelload, analizseries, bootstrep. Розміщує часові ряди на компоненту TStringGrid, зберігає обрані ряди та керує Excel об'єктом. Викликає допоміжні вікна.

Клас bootstrep являється допоміжним вікном програмного комплексу, щодня необхідне для побудови агрегованих ЧР, класифікації АЧР та його виклику.

Клас analizseries (Форма завантаження ЧР) являється допоміжним вікном програмного комплексу, необхідне для введення параметрів методу побудови АЧР часового ряду та його виклику.

Клас ftsms являється допоміжним вікном програмного комплексу, необхідне для введення параметрів форм представлення ЧР, забезпечення їх виклику.

Клас `ftschart` являється допоміжним вікно програмного комплексу, необхідне для визначення форми побудови прогнозу та інтерполяції елементів ЧР, а також відображення графіків в декількох режимах.

Клас `editftsms` являється допоміжним вікно програмного комплексу, необхідне для введення форм аналізу та оцінювання моделей ЧР, а також додання рядів на графіки.

Клас `exelload` являється допоміжним вікно програмного комплексу, необхідне для вибору варіантів моделей АЧР, а також для завантаження даних з Excel.

Опис класів зображено на рис. 3.3.

Main TStringGrid *SG; int left; int right; int top; int bottom; Variant app; void FormCreate(TObject *Sender); void load(TObject *Sender); void save(TObject *Sender); void graphic(TObject *Sender); void buildfts(TObject *Sender); void analizser(TObject *Sender); void buildbootstrep(TObject *Sender);	analizseries TEdit *count; TLabel*Label1; TButton *Build; void buildanalseri(TObject *Sender);	exelload TButton *load; TButton *cancel; void loadClick(TObject *Sender); void cancelClick(TObject *Sender);
ftsms TLabel *Label3; TEdit *forepoint; TButton *Build; TEdit *kappa; TLabel *Label5; void buildfts(); void analizseries(int coun); void bootstrep(int fiks, int count);	bootstrep TEdit *countseries; TLabel*Label1; TButton *build; void builbootstrep(TObject *Sender);	
	MultiSten float kappa; float forecast; MultiStep(); Void Calculate(float x1,float x2,float x3, float kappa); float Getforecast();	editftschart TEdit *namechart; TLabel *Label1; void addchart(TObject *Sender);
DateConvert TDate *mdate; int *mstep; int size; DateConvert(TStringGrid *SG,int Bottom,int Top,int left); DateConvert(TStringGrid *SG,int Bottom,int Top,int index,UnicodeString start); int Getstep(int index);	ftschart TChart *Chart1; TMenuItem *view; TMenuItem *value; TMenuItem *valuetime; TMenuItem *separabel; int posseries[8]; void analizseries(); void clearchart(); void valueview(TObject *Sender); void valuetimeviewClick(TObject *Sender); void separabelview(TObject *Sender);	

Рисунок 3.3 – Класи програмного комплексу

Клас DateConvert необхідний для конвертації дати

Клас Multistep необхідний для розрахунків за формама аналізу та оцінювання моделей ЧР .

Зв'язок між класами демонструє рис. 3.4.

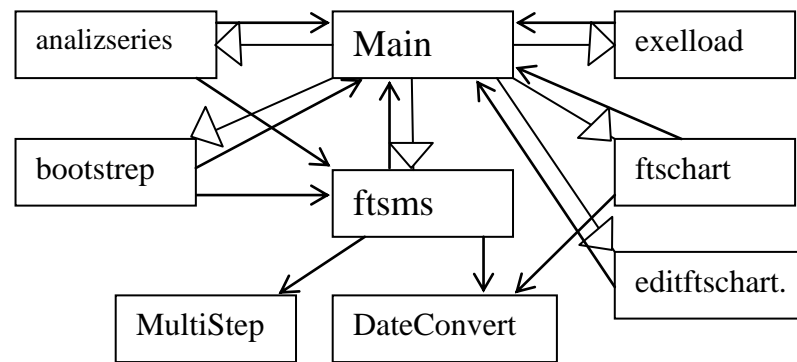


Рисунок 3.4 – Основні зв'язки між класами

3.3.3. Принципи проектування внутрішньої структури системи

При проектуванні внутрішньої структури системи використовувалися такі загальні принципи об'єктно-орієнтованого проектування:

- забезпечення слабкої зв'язності об'єктів що взаємодіють. За ним чим менше об'єкти знають один про одного, тим гнучкіше створювана програмна система. Кожному компоненту немає необхідності «знати» внутрішній устрій іншого;

- забезпечення взаємодії тільки з «близькими» компонентами. Принцип спрямований на мінімальну інформованість – при проектуванні класу треба звертати увагу на кількість класів, з якими буде відбуватися взаємодія. Чим менше таких класів, тим гнучкіше система;

- голлівудський принцип – не викликайте нас, ми самі вас викличемо. За Фаулера - це синонім принципу IoC. Згідно ньому компоненти високого рівня (наприклад, інтерфейси) визначають за компоненти низького рівня (реалізації), як і коли їм підключатися до

системи. Інші автори Head First Design Patterns допускають, що за цим принципом компоненти низького рівня можуть приймати участь в обчисленнях без формування залежностей з компонентами високого рівня. Саме в цьому полягає відмінність такого проектування від більш жорсткого принципу IoC;

- розділення інтерфейсу, за яким перевага віддається тому проекту, коли у системі використовується велика кількість спеціалізованих інтерфейсів замість кількох універсальних.

3.4.Формування інтерфейсу користувача програмного комплексу

При проектуванні інтерфейсу застосовувався принцип інтуїтивної зрозумілості, при рцьому інтерфейс був не нагромаджений великою кількістю елементів. Для цього було стпорено кілька форм, кожна з яких виконує свою роль.

Головна форма програмного комплексу зображена на рис. 3.5

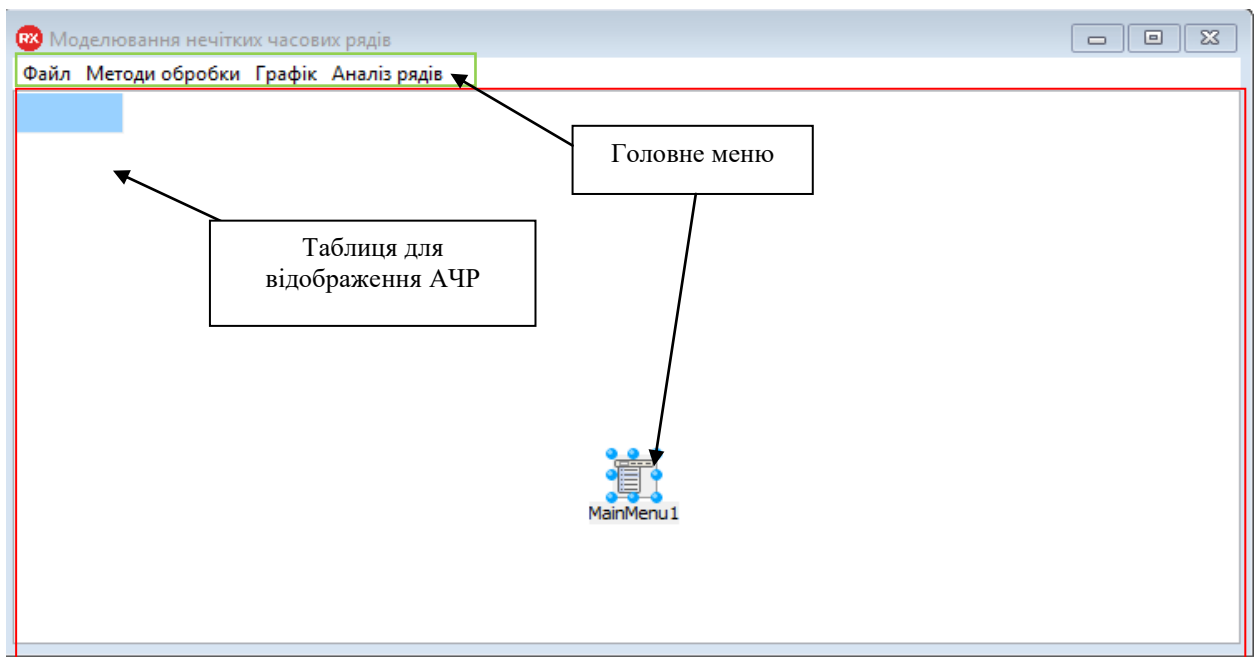


Рисунок 3.5 – Головне вікно програмного комплексу дослідження моделей антиперсистентних часових послідовностей

Вікно меню, призначене для введення параметрів методів аналізу АЧР, зображено на рис. 3.6.

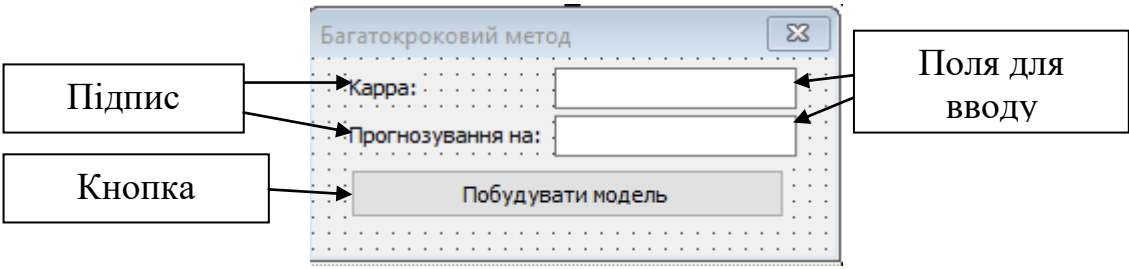


Рисунок 3.6 – Допоміжне вікно для введення параметрів методів аналізу АЧР

Вікно для введення параметрів даних АЧР, призначених для процедур класифікації, зображене на рис. 3.7.

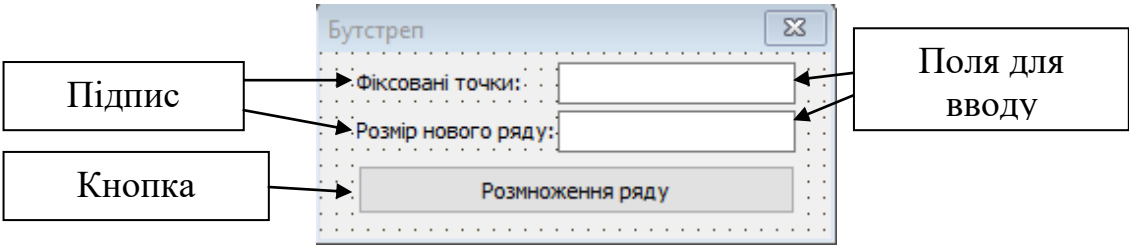


Рисунок 3.7 – Вікно для параметрів методу класифікації АЧР

Допоміжне вікно для введення параметрів процедури вибору варіантів агрегування для моделей ЧР зображене на рис. 3.8.

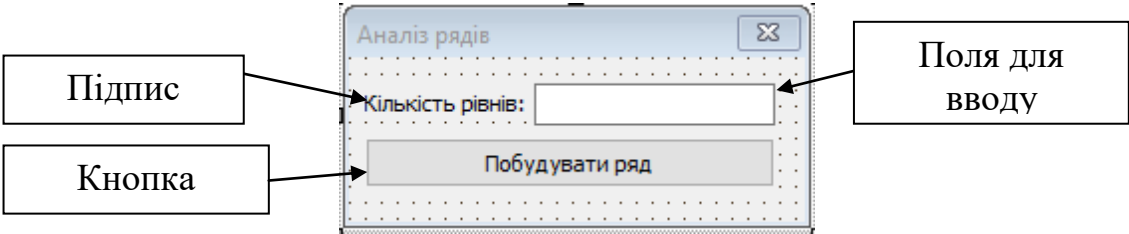


Рисунок 3.8 – Допоміжне вікно для побудови ряду на основі агрегування

Вікно для завантаження даних НДП з Microsoft Office Excel зображено на рис. 3.9.

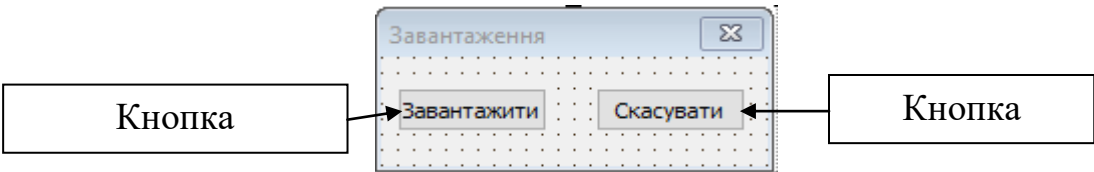


Рисунок 3.9 – Вікно для завантаження АЧР з Microsoft Office Excel

Допоміжне вікно для формування графіку АЧР зображено на рис. 3.10.

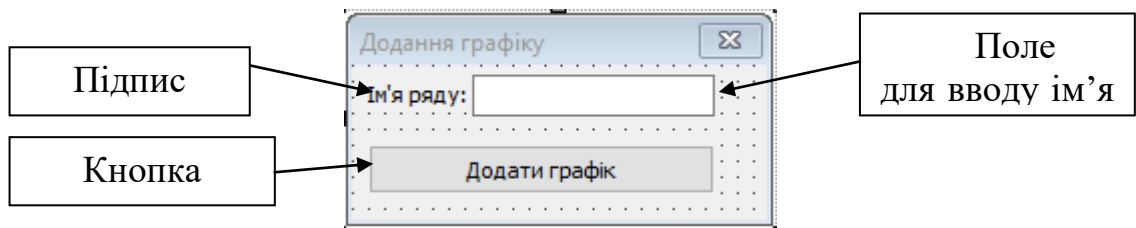


Рисунок 3.10 – Вікно для формування графіку АЧР

Вікно для відображення ряду на графіку зображено на рис. 3.11.

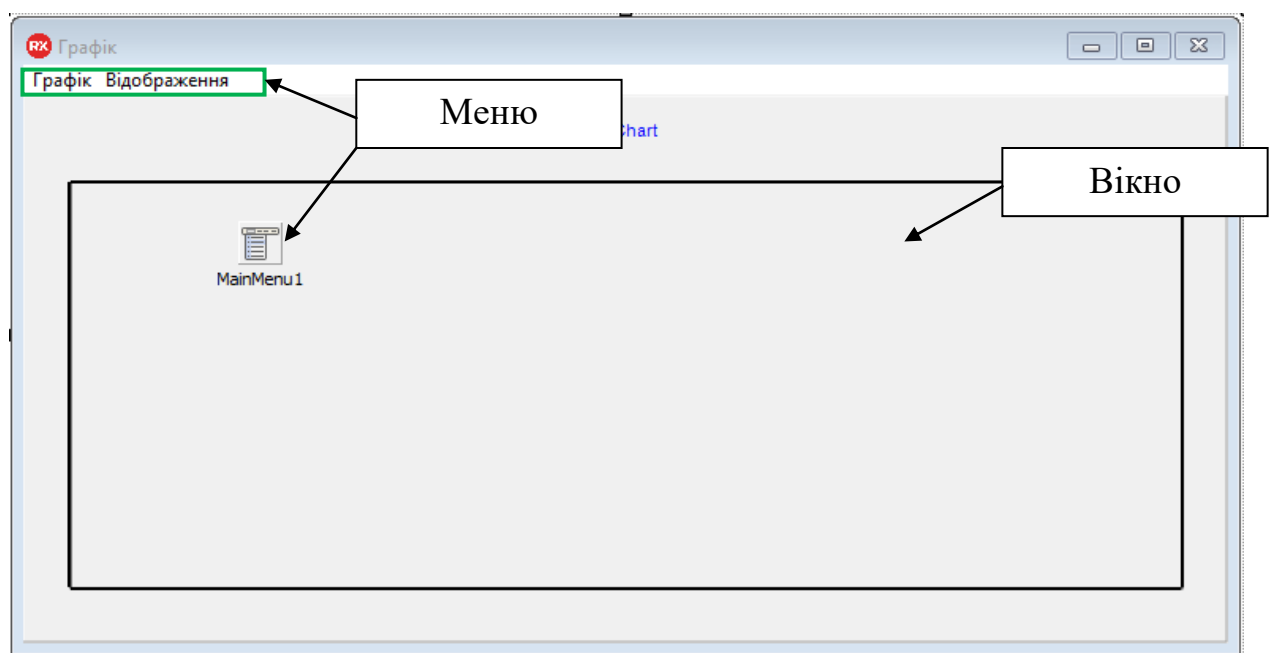


Рисунок 3.11 – Вікно для відображення графіку ЧР

3.5. Тестування та налагодження програми

3.5.1. Аналіз методів тестування та налагодження програм

Етапи тестування та відлагодження програм є найбільш трудомісткими в процесах розробки програм. Мета тестування – це виявлення помилок у роботі програмного комплексу. Виявлення синтаксичних помилок в більшості виконується за допомогою компілятора, вони можуть виправляються ще на момент написання. Для виявлення логічних помилок, помилок типізації, помилок узгодженості параметрів площндур та ін. можна отримати лише дослідивши роботу алгоритму, шляхом порівняння

очікуваних результати роботи з спеціально сформованими вихідними даними щодо роботи алгоритму.

При виявленні логічної помилки її необхідно локалізувати та виправити. Для цього на практиці використовують процедури налагодження, коли знають в якому алгоритмі знаходиться локалізована логічна помилка. При відлагодженні перевіряється робота (виконання та дані) кожної команди, та команда алгоритму, яка не відповідає встановленим вимогам, і буде розглядатися як можлива логічна помилка..

Загальними та ефективними методами тестування алгоритму являються метод «чорної скриньки» та метод «білої скриньки».

При застосуванні методу «чорної скриньки» полягають що алгоритм певної функції невідомий, а відомі лише вхідні та очікувані вихідні данні функції, що підходить для виявлення логічної помилки у функції.

Метод «білої скриньки» передбачає що алгоритм реалізації функції відомий, а система вхідних даних будуються так, щоб покрити максимальну кількість гілок виконання алгоритму. Таким чином виявляється ділянка алгоритму, де можливо знаходиться логічна помилка. Після виявлення помилки починається етап відлагодження для виправлення помилки, з подальшим повторним тестуванням.

Для прикладу розглянемо специфікації та текст функції `buildfts`

Атрибути функції: - вхідна змінна, обраного верхнього індексу; - вхідна змінна, обраного нижнього індексу; - вхідна змінна, обраного лівого індексу; - вхідна змінна, обраного правого індексу; - вхідна змінна, параметру «каппа»; - вхідна змінна, параметру кількість точок для прогнозу; - вхідна таблиця що зберігає двовимірний масив значень нечіткої часової послідовності.

Сутність виконання функції полягає у тому, що її виконання повертає одномірний або двовимірний масив, в якому знаходиться розрахована модель. У разі коли індекси, або зміни параметрів були введені не коректно, тоді функція повертає порожній масив даних.

```

// Моделювання прогнозу
Float* TForm2::buildfts(int top, int bottom,
int left, int right, float kappa, int forepoint, TStringGrid SG)
{
//перевірка параметрів моделі
if(forepoint->Text>2)
if(kappa->Text!=NULL)
{
//створення масиву, необхідного для збереження даних розрахованої
моделі,
    float *mas=new float[bottom-top+forepoint];
    //виконання резервування даних для розрахунку, кодування та
декодування
    DateConvert *dconvert,*ddecode;
    MultiStep* forecast=new MultiStep[bottom-top];
    MultiStep* fors;
}
// перевірка розмірності
If(mas!=NULL)
    if(right==left)
    {
        //одномірний масив
//перевірка умови для оперування датою
        bool flag=false;
// перевірка чи є масив значеннями типу дата
        if(SG->Cells[right][top].Pos(".")!=0)
        {
// конвертування дати до чисел
            flag=true;
            dconvert=new DateConvert(SG, bottom+1,top,left);
            for(int x=0;x<=bottom-top+1;x++)
                mas[x]=dconvert->Getstep(x);
        }
        else

```

```

        {
// запис даних без перетворень
        for(int x=top;x<=bottom;x++)
            mas[x-top]=StrToFloat(SG->Cells[right][x]);
        }

// виконання розрахунків
fors=calculeteforecast(mas,bottom-top+forepoint,
forepoint,kappa);
// запис даних до масиву
for(int x=top+3;x<=bottom+forepoint+1;x++)
{
    mas[x]=fors[x-top-3].Getforecast();
}

//перевірка на «дата» та виконання декодування
    if(flag)
    {
//декодування значень в дату
ddecode=new DateConvert(SG,bottom+forepoint+1,top+3,
SG->ColCount-2,
SG->Cells[right][top+2]);

        for(int x=top+3;x<=bottom+Strforepoint+1;x++)
            SG->Cells[SG->ColCount-1][x]=
ddecode->Getdate(x-top-3);
//видалення тимчасових змінних
            delete ddecode;
            delete dconvert;
        }
    }
    else
    {
//конвертування типу «дата» до чисел, запис результатів

        dconvert=new DateConvert(SG, bottom+1,top,left);

```

```

        for(int x=0;x<= bottom- top+1;x++)
            mas[x]=dconvert->Getstep(x);
//Виконання розрахунків
fors=calculeteforecast(mas, bottom-top+forepoint,
forepoint,kappa);
        for(int x=top+3;x<bottom+forepoint+1;x++)
        {
            SG->Cells[SG->ColCount-3][x]=
FloatToStr(fors[x-top-3].Getforecast());
        }
// конвертування даних «даа» до чисел, запис результату
        ddecode=new DateConvert(SG,bottom+ forepoint+1,
top+3,SG->ColCount-3,SG->Cells[left][top+2]);
//запис декодованої дати
        for(int x=top+3;x<bottom+forepoint+1;x++)
            SG->Cells[SG->ColCount-2][x]=
ddecode->Getdate(x-top-3);
//запис розрахованих значень ряду
        for(int x=top;x<=bottom;x++)
            mas[x-top]=StrToFloat(SG->Cells[right][x]);
//Виконання розрахунки щодо отримання значень
fors=calculeteforecast(mas, bottom-top+ forepoint,
forepoint,kappa);
// перенесення результатів розрахунків
        for(int x=top+3;x<bottom+forepoint+1;x++)
        {
            SG->Cells[SG->ColCount-1][x]=FloatToStr(fors[x-top-
3].Getforecast());
        }
    }
//звільнення пам'яті
    delete forecast;
    if(fors!=NULL)
        delete fors;

```

```

}
//повернення даних розрахованої моделі.
return mas;
}
//-----

```

На рис. 3.14 представлена узагальнена блок-схема функції що тестується buildfts.

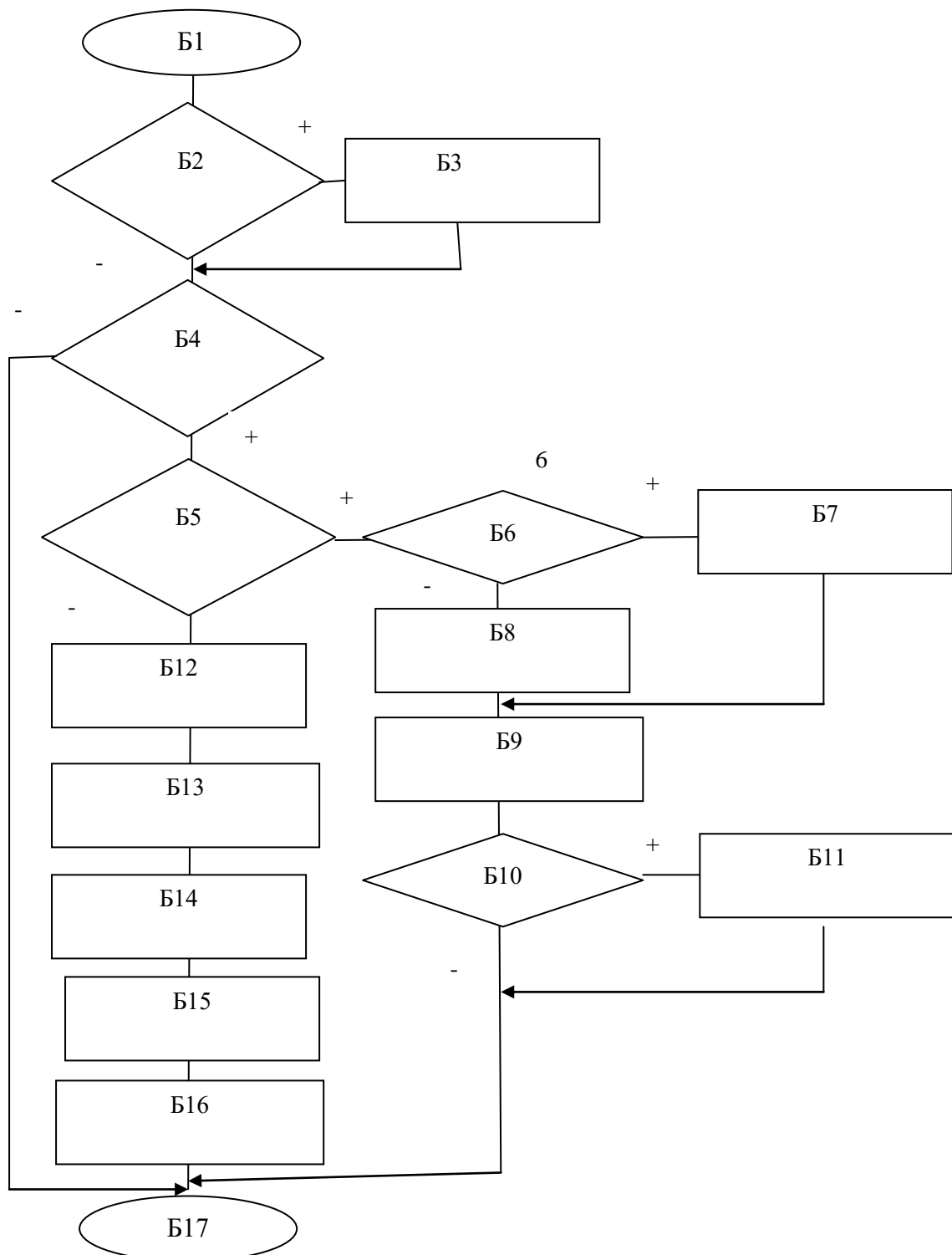


Рисунок 3.12 – Блок-схема функції оперування змінними типу «дата»,
формування buildfts

На рисунку блоки визначають наступне: Б1 – Початок, Б2 – Перевірка параметрів моделі, Б3 - Створення масиву призначеного для розрахунків, Б4 – визначення умови «Масив створений», Б5 – Визначається чи є розмірність масиву1, Б6 – Масив дати , Б7 - Декодування дати до цілих чисел, Б8 - Записуються масив значення, Б9 – Розрахунок та запис даних моделі процесу, Б10 – Перевірка умови «якщо дата», Б11 - Декодування цілих чисел до формату дати, Б12 - Декодування дати та перетворення до цілих чисел, Б13 – Розрахунок моделі даних дати, Б14 - Декодування послідовностей цілих чисел до форматів дати. Б15 – Записують масив значень, Б16 – Процедура декодування цілих чисел до формату дати. Б17 - Кінець.

Для тестування функції buildfts оперування даними типу «дата» розроблені тести, представлені у табл. 3.1.

Таблиця 3.1

Варіанти тестів до функції buildfts рис. 3.12

Номер тесту	Вхідні значення				Очікувані вихідні значення
	Масив значень	Верхній, нижній, лівий, правий індекси	Параметр алгоритму «каппа»	Кількість точок прогнозу	
1	[5, 2.4, 9, 7, 5]	5, 1, 1,1	0	0	NULL
2	[5, 2.4, 9, 7, 5]	5, 1, 1,1	0	3	[4.5, 4.9, 7.5, 6.3, 5.8]
3	[27.12.2014, 15.03.2015, 20.06.2015,	5, 1, 2,2	0	3	[05.08.2015, 04.10.2015, 01.12.2015, 23.12.2015, 11.01.2016]

Таблица 3.2

[illegible]

Висновки до розділу 3

В результаті розробок за розділом виконана формалізація процесів програмного комплексу «Дослідження моделей антиперсистентних часових послідовностей» у вигляді діаграми варіантів. Діаграма варіантів представляє всі завдання, які необхідно реалізувати. Для формування базової архітектури програми використана архітектура бібліотеки VCI.

Розроблено модель для відображення взаємодія між частинами архітектури програмного комплексу «Дослідження моделей антиперсистентних часових послідовностей», як взаємодія головних елементів архітектурної моделі VCL

В якості засобів програмної реалізації обране середовище для розробки RAD Studio Builder. Таке рішення ґрунтується на тому, що завдяки бібліотеці візуальних компонентів проектування інтерфейсу розробка програми виконуються швидко та ефективно. Вибір мови реалізації C++ зроблено через те, що компоненти середовища RAD Studio Builder написані на C++, Такі засоби розробки програми є доцільними через велику кількість готових бібліотек та можливостей мови C++. У розділі засобами UML діаграм описані класи моделі програми, а також їхня взаємодія.

У розділі наведено елементи щодо опису процесу проектування інтерфейсу користувача. Для розробки інтерфейсу використовувалися знайомі користувачу елементи; завдяки чому користувач не буде перевантажений громіздким інтерфейсом. Виконане тестування елементу розробленого інструментального забезпечення для моделювання часових рядів, ефективності алгоритмів перевірено за допомогою методів «білої скриньки» та методу «чорної скриньки». Запропоновані та застосовані процедури розробки програмного забезпечення дозволяє зробити висновок про забезпечення достатньої якості та достовірність результатів досліджень.

4. ДОСЛІДЖЕННЯ МЕТОДУ МОДЕЛЮВАННЯ ТА ПРОГНОЗУВАННЯ АНТИПЕРСИСТЕНТНИХ ЧАСОВИХ ПОСЛІДОВНОСТЕЙ.

4.1. Підготовка до чисельного експерименту

4.1.1. Опис використаного програмно-апаратного середовища

Методи моделювання та прогнозування АЧР були досліджено за допомогою розробленого програмного комплексу «Дослідження моделей антиперсистентних часових послідовностей». На рис. 4.1 зображено функціональні можливості програмного комплексу. Для перевірки достовірності результатів з використанням подібних методів програмних аналогів немає.

Приведений на рис. 4.1 програмний комплекс реалізує наступні функції.

- 1) управління даними (уведення ЧР, ведення ЧР, видалення ЧР, Уведення даних моделей полігонів $f(x,a)$).
- 2) відображення даних: (графік ЧР, таблиця ЧР, відображення ЧР, відображення даних аналізу ЧР).
- 3) аналіз даних ЧР моделювання, формування ЧР моделі, розрахунок коефіцієнта Херста, розрахунок коефіцієнта детермінації, формування середніх рівня, класифікація ЧР, формування моделі полігону, процедури інтерполяції, оцінка похибки моделювання.

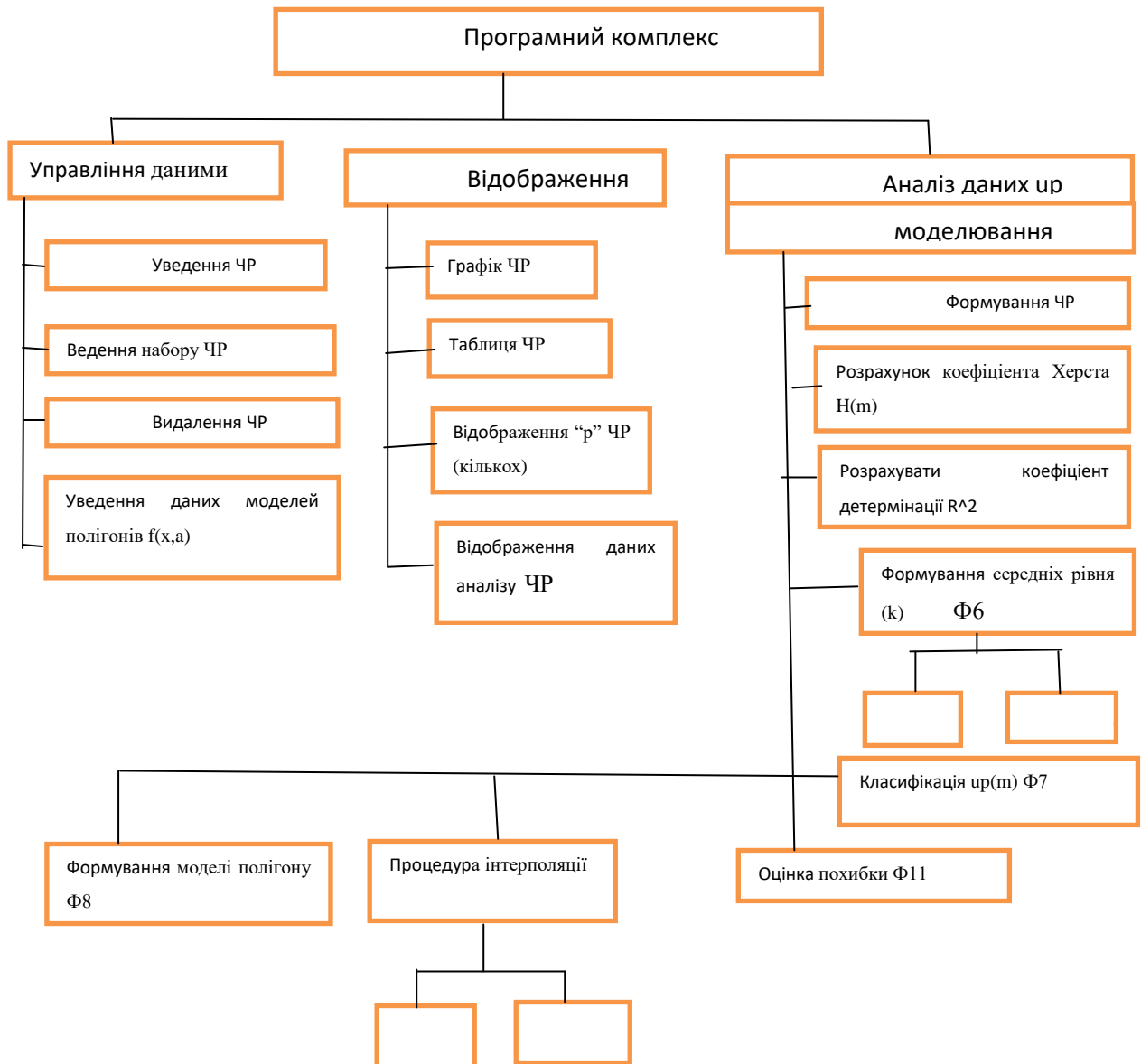


Рисунок 4.1 – Функціональні можливості програмного комплексу

Програмний комплекс класифікація:

- 1) управління даними, введення ЧР, ведення ЧР, видалення ЧР, введення даних моделей полігонів,
- 2) відображення даних: графік ЧР, таблиця ЧР, відображення ЧР, відображення даних аналізу ЧР,
- 3) аналіз даних ЧР та моделювання, формування ЧР моделі, розрахунок коефіцієнта Херста, розрахунок коефіцієнта детермінації, формування середніх рівня, класифікація ЧР, формування моделі полігону, процедури інтерполяції, оцінка похибки

Дослідження проводились з використанням наступного апаратного середовища:

- процесором AMD Ryzen 2200g з тактовою частотою 3.5 ГГц;
- об'єм оперативної пам'яті 16 Гб; об'єм жорсткого диску 1Тб;
- дискретна відео карта Radeon 5700 XT;

4.1.2. Опис методу моделювання часових послідовностей

Для дослідження загальних властивостей ЧР-В використані методи фрактального аналізу, обчислення показника Херста. Його значення дозволяють встановити категорію, класифікувати процеси ЧР на персистентні (трендостійкі), антиперсистентні (злам тенденції, повернення), виявити випадковий характер процесу.

$$H = \frac{\log\left(\frac{R}{S}\right)}{\log(a * N)} \quad (4.1)$$

де : H – показник Херста; S – середнє квадратичне відхилення ряду спостережень; R – розмах накопиченого відхилення; N – число періодів спостережень; a – задана константа.

4.1.3. Опис методу агрегування ЧР

Структура схем варіантів агрегування АЧР виявилася дуже суттєвою для завдання утворення класифікації АЧР. Не при всіх схемах агрегування рівнів вихідних рядів удалося отримати стійкі класи, що характеризують процеси, дозволяють їх відрізнити сфіж собою. На рис. 4.2 приведені приклади варіантів схем агрегування півнів АЧР при утворення модельних ЧР, які далі застосовують для ьрозрахунків параметрів Херста, то ж і для визначення можливого класу АЧР. Клас визначався як найменший

можливий набір рівнів агрегування, при якому виконувалася умова виходу АЧР із категорії антиперсистентних.

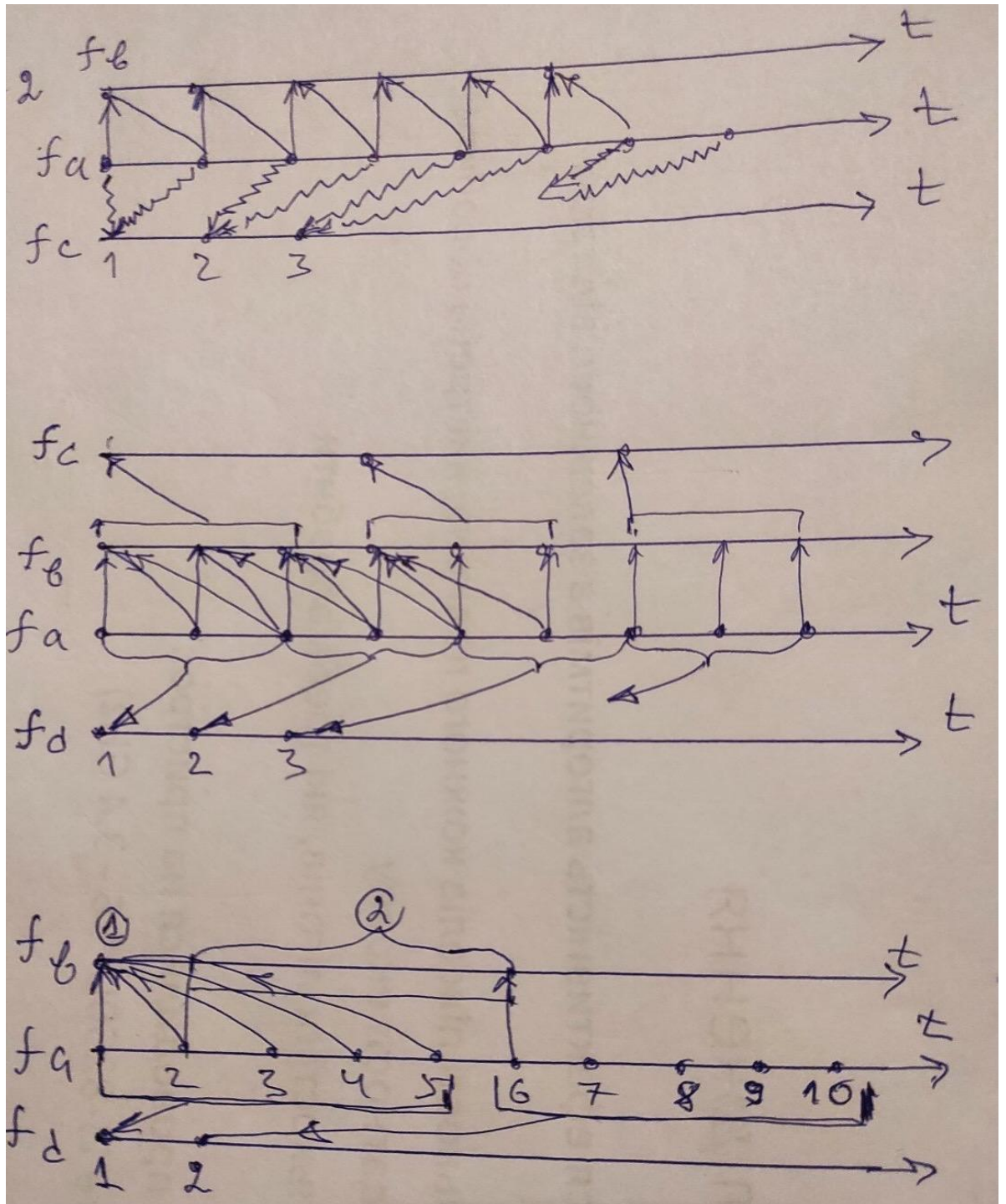


Рисунок 4.2. Схеми варіантів утворення агрегованих ЧР для різних типів класифікації АЧР

4.2. Опис методу побудови інтерполяції часових послідовностей

Приведені результати досліджень щодо класифікації та прогнозування АЧР, що отримані з використанням різних варіантів схем агрегування, та розширеної структури системи, подана на рис. 4.3. В ній для дослідження моделей АЧР додані нові функції аналізу. Укажимо нові модулі системи: блок вибору варіантів моделей схем агрегування (М1/М2), блок формування агрегованих, узагальнених часових послідовностей відповідних типів Б4, модуль формування та дослідження математичних моделей трендів утворених узагальнених ЧР Б6. Зазначений модуль програмної системи виконує розрахунки прогнозних показників моделей процесів, а також інтерполяцію внутрішніх рівнів, використовуючи показники що характеризують діапазон у цілому.

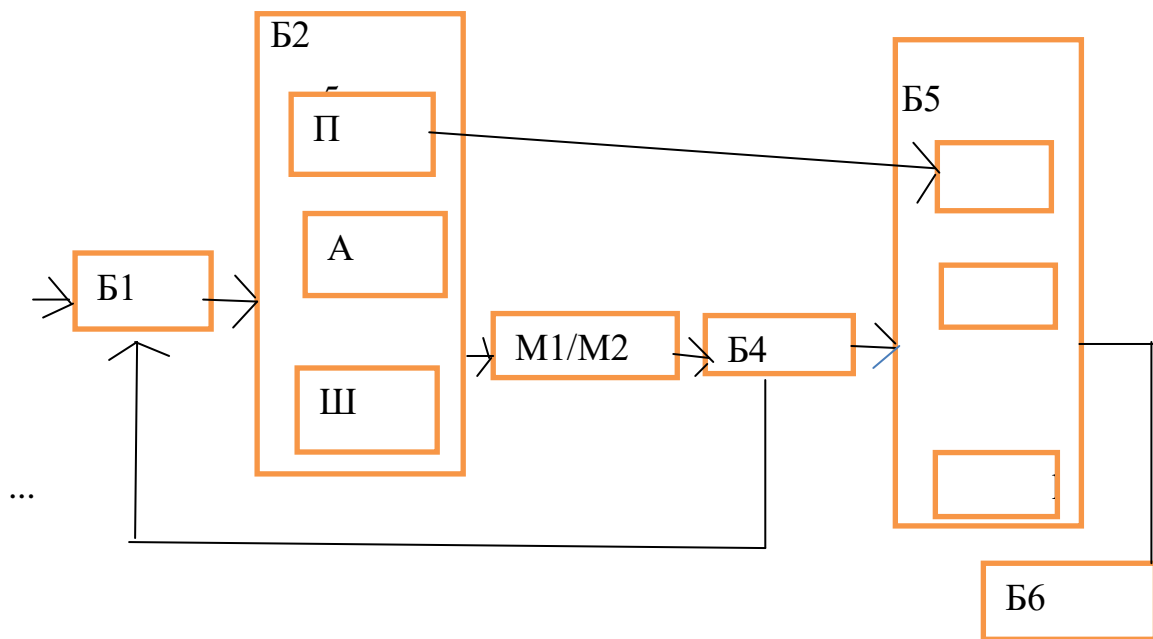


Рисунок 4.3. Розширена структура системи дослідження на прогнозування показників АЧР.

Необхідні для реалізації процедур інтерполяції величин рівнів АЧР схематично зображені на рис. 2.7. При цьому вони демонструють граничні варіанти можливих схем утворення агрегованих послідовностей даних для визначення 5-го класу антиперсистентного ЧР. Це модель позначена як f_b ,

коли агреговані моделі містять послідовні члени, що відрізняються одним елементом, а саме

$$u_{b1} = (u_{a1} + u_{a2} + u_{a3} + u_{a4} + u_{a5})/5;$$

$$u_{b2} = (u_{a2} + u_{a3} + u_{a4} + u_{a5} + u_{a6})/5;$$

$$u_{b3} = (u_{a3} + u_{a4} + u_{a5} + u_{a6} + u_{a7})/5;$$

...

Для формування протилежної схеми агрегування АЧР та утворення нових послідовностей, коли послідовні елементи агрегованих моделей не містять попередні елементи, визначається інша система рівнянь, тобто

$$u_{d1} = (u_{a1} + u_{a2} + u_{a3} + u_{a4} + u_{a5})/5;$$

$$u_{d2} = (u_{a6} + u_{a7} + u_{a8} + u_{a9} + u_{a10})/5;$$

$$u_{d3} = (u_{a11} + u_{a12} + u_{a13} + u_{a14} + u_{a15})/5;$$

...

Для розрахунків значень вузлів інтерполяції використані дані та рівняння, які подібні до наведених на рис. 2.5 – рис. 2.7. Такі приклади варіантів агрегування даних були використані для побудови узагальнених коректних математичних моделей недетермінованих процесів, сформованих за даними спостережень. Результати розрахунків, подані далі у розділі, демонструють змістовність і широкі можливості утворення агрегованих моделей процесів у вигляді ЧР. В проведеному дослідженні було встановлено головне – при яких формах агрегування забезпечується можливість класифікації вихідних ЧР. Наступне, результати дозволили запропонувати ефективні алгоритми прогнозування за агрегованими моделями. Такі алгоритми також дозволяють виконувати інтерполяцію та розраховувати внутрішні рівні показників процесів ЧР, використовуючи оцінки інтегрованих величин, отриманих за моделями трендів виду (f_b, f_c) та подібних до них.

В експериментальних розрахунках для різних варіантів агрегування АЧР були застосовані окремі процедури інтерполяції, які чисельно ефективно давали розрахунки величин внутрішніх рівнів на основі прогнозних величин, визначених для всього діапазону агрегування, та отриманих на основі системи рівнянь (2.2). Приклад моделі узгоджених рівнянь, необхідних для виконання одного кроку процедури інтерполяції, наведено нижче. При цьому для класу АЧР ($k=3$) за отриманими моделями f_c для розрахування внутрішньо інтервальних значень прогнозних рівнів замість рішення системи рівнянь виду (2.2) були використані $(k-1)$ початкові значення АЧР ряду Y . За рахунок цього для розрахунку значень інтерполяції рівнів необхідно використовувати такі співвідношення

$$\begin{aligned} Y_1 &= \frac{y_1 + y_2 + y_3}{3}, \text{ то } \rightarrow \hat{y}_3 = 3 * Y_1 - y_1 - y_2 \\ Y_2 &= \frac{y_2 + y_3 + y_4}{3}, \text{ то } \rightarrow \hat{y}_4 = 3 * Y_2 - y_2 - \hat{y}_3 \\ Y_3 &= \frac{y_3 + y_4 + y_5}{3}, \text{ то } \rightarrow \hat{y}_5 = 3 * Y_3 - \hat{y}_3 - \hat{y}_4 \end{aligned}$$

замість рішення систем рівнянь. При відомих оцінках прогнозованного значення Y_1 та заданих значеннях (y_1, y_2) можливо розрахувати y_3 . При відомих оцінках прогнозованного значення Y_2 та заданих значеннях (y_2, y_3) можливо розрахувати y_4 і далі за такою схемою. Разом зтим для реалізації зазначеної ланцюгової форми алгоритмів інтерполяції була утворена відповідна структура програмної реалізації, яка забезпечує утворення модельної послідовності що містить всі попередні рівні ЧР.

4.3. Проведення експериментів з класифікації АЧР

В експериментальних розрахунках значень показників інтерполяції рівнів, отриманих за моделями процесів, в дослідженнях алгоритмів використовувалися $(k-1)$ початкове значення ряду Y , що відповідає прикладу системи рівнянь для класу АЧР ($k=3$) вмду:

$$Y_1 = \frac{y_1 + y_2 + y_3}{3}, \text{ то } \rightarrow \hat{y}_3 = 3 * Y_1 - y_1 - y_2$$

$$Y_2 = \frac{y_2 + y_3 + y_4}{3}, \text{ то } \rightarrow \hat{y}_4 = 3 * Y_2 - y_2 - \hat{y}_3$$

$$Y_3 = \frac{y_3 + y_4 + y_5}{3}, \text{ то } \rightarrow \hat{y}_5 = 3 * Y_3 - \hat{y}_3 - \hat{y}_4$$

Експерименти з дослідження методу класифікації проводились за допомогою застосування розробленого програмного комплексу «Дослідження моделей антиперсистентних часових послідовностей», яка дозволяє прогнозувати значення антиперсистентної часової послідовності.

При цьому досліджувалися різні схеми утворення гарегованих моделей, відповідно рис. 4.2. На рис. 4.4 представлено результати моделювання при дослідженні класу ($k=3$), коли схеми були з перекриттям рівнів при агрегуванні. На рис. 4.5 представлено результати моделювання при дослідженні класу ($k=3$), коли схеми були без перекриття 2-го рівня, див. рис. 4.2.

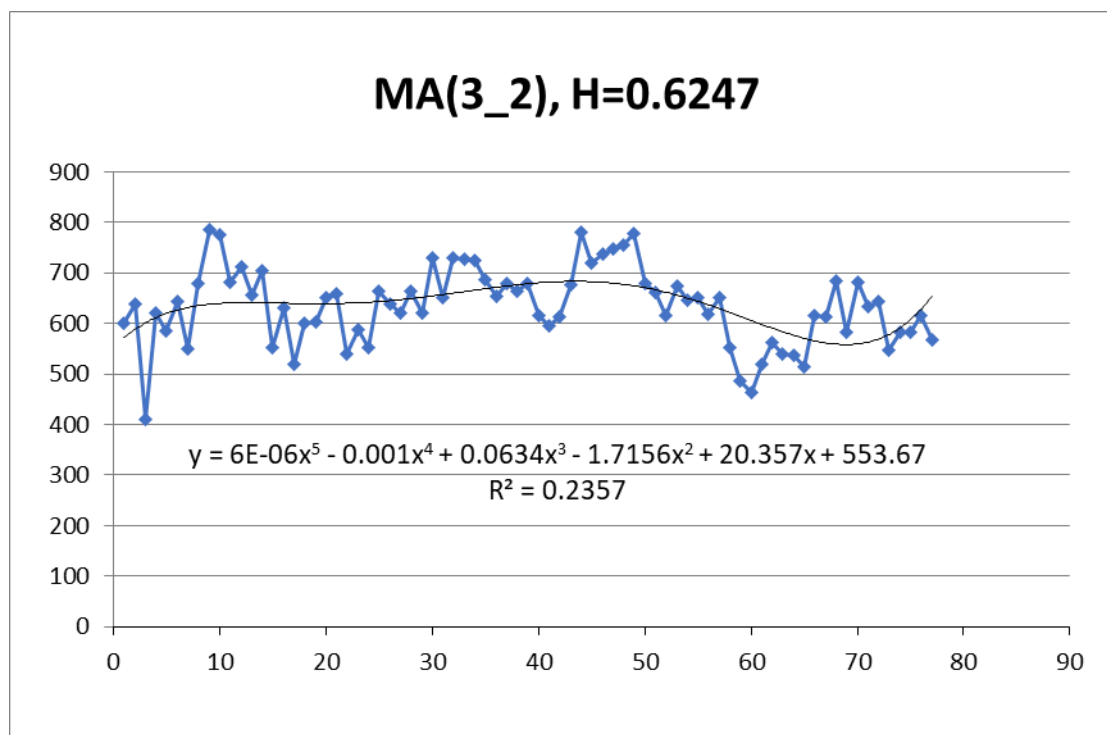


Рисунок 4.4. Агрегування по ($k=3$): перекриттям рівнів при агрегуванні

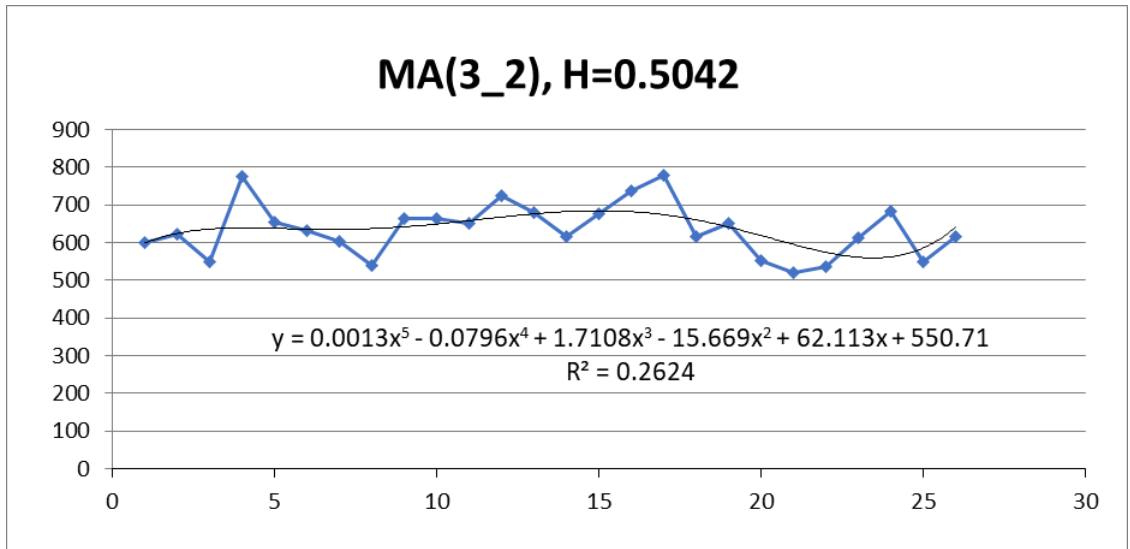


Рисунок 4.5. Агрегування по класу ($k=3$): без перекриття 2-го рівня

На рис. 4.6 представлено результати моделювання при дослідженні класу ($k=3$), коли схеми були з перекриттям 3 рівня при агрегуванні. На рис. 4.7 представлено результати моделювання при дослідженні класу ($k=3$), коли схеми були без перекриття 3-го рівня.

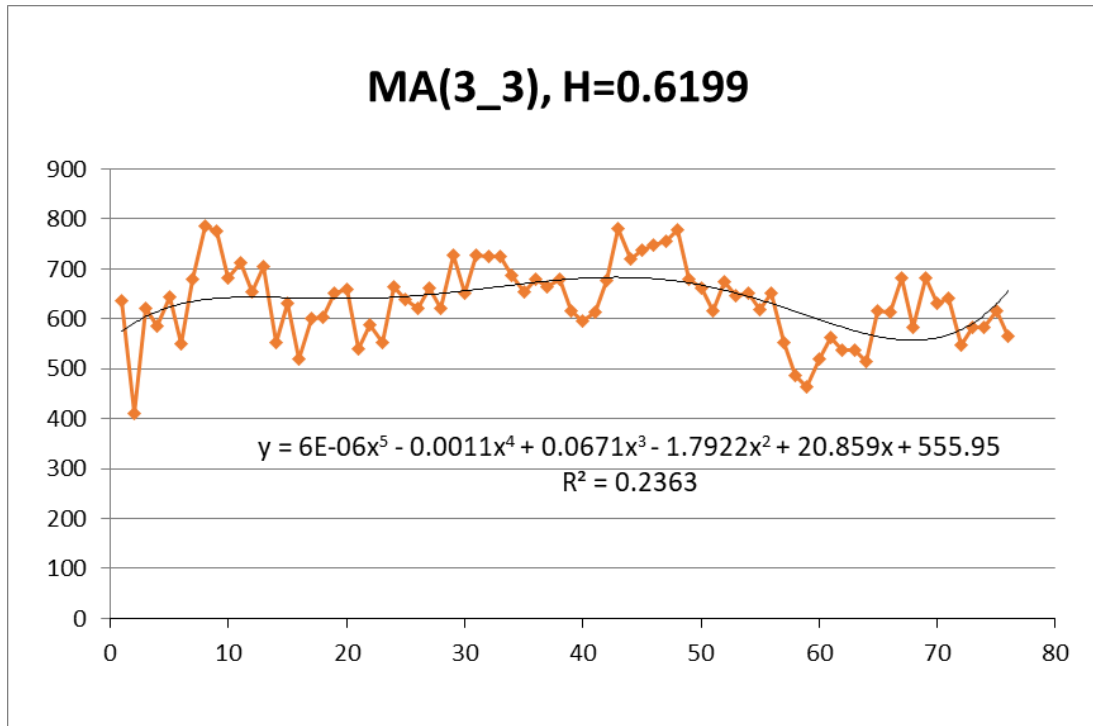


Рисунок 4.6. Агрегування по ($k=3$): з перекриттям 3 рівня при агрегуванні

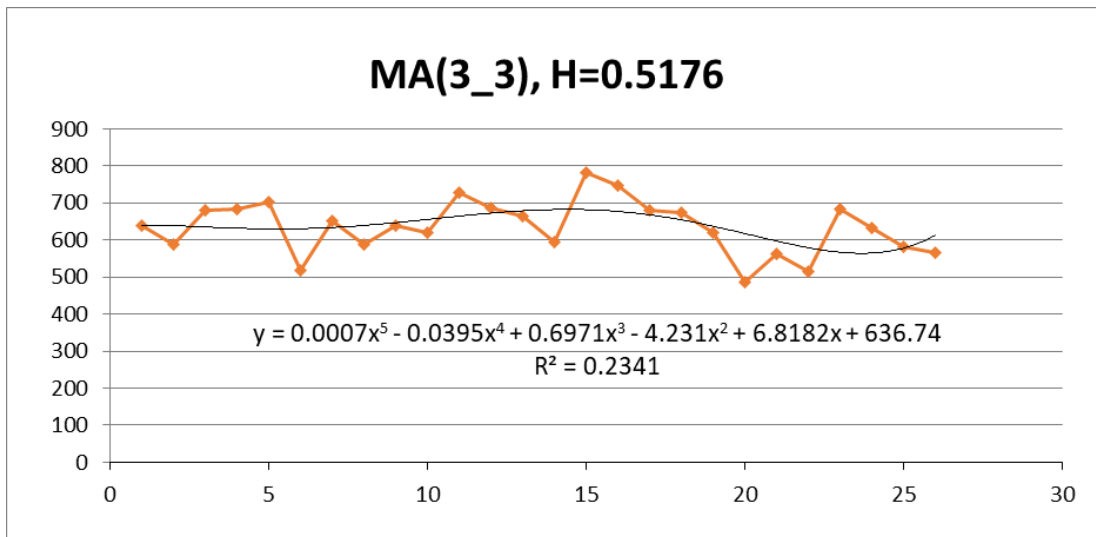


Рисунок 4.7. Агрегування по ($k=3$): без перекриття 3-го рівня

Рис.4.4 – рис. 4.7 містить графіки . ЧР отримані в результаті агрегації по класу ($k=3$) рівнів з перекриттям з відповідними моделями та оцінкою достовірності (R^2).

При побудові прогнозу показника антиперсистентного процесу використовується наступний алгоритм:

1. Для Y_1 вибирається k -рівнів початкового ЧР, починаючи з $i=1, n$ по k . Для Y_2 – $i=2, n$ по k , і тд, для всіх Y_j , $j=1, k$

$$Y_i = \frac{\sum_{i=1}^k y_i}{k}$$

2. Визначається коефіцієнт Херста, будується прогноз та розраховується коефіцієнт детермінації (R^2).
3. За отриманими моделями розраховуються внутрішньо інтервальні значення прогнозних рівнів, для визначення яких використовуються k - l початкові значення ряду Y . Наприклад, для ($k=3$):

-

$$Y_1 = \frac{y_1 + y_2 + y_3}{3}, \text{ то } \rightarrow \hat{y}_3 = 3 * Y_1 - y_1 - y_2$$

$$Y_2 = \frac{y_2 + y_3 + y_4}{3}, \text{ то } \rightarrow \hat{y}_4 = 3 * Y_2 - y_2 - \hat{y}_3$$

$$Y_3 = \frac{y_3 + y_4 + y_5}{3}, \text{ то } \rightarrow \widehat{y}_5 = 3 * Y_3 - \widehat{y}_3 - \widehat{y}_4$$

4. Таким самим чином розраховується всі наступні рівні прогнозного ряду (рис.4.8).

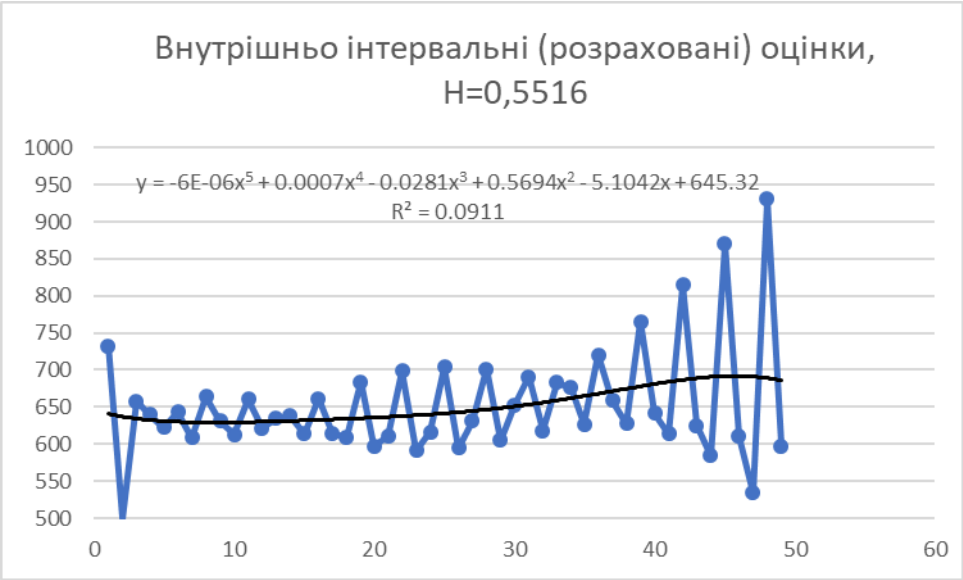


Рисунок 4.8. Внутрішньо інтервальні рівні, побудовані на основі моделей MA3_1 – MA3_3 без перекриття рівнів агрегування

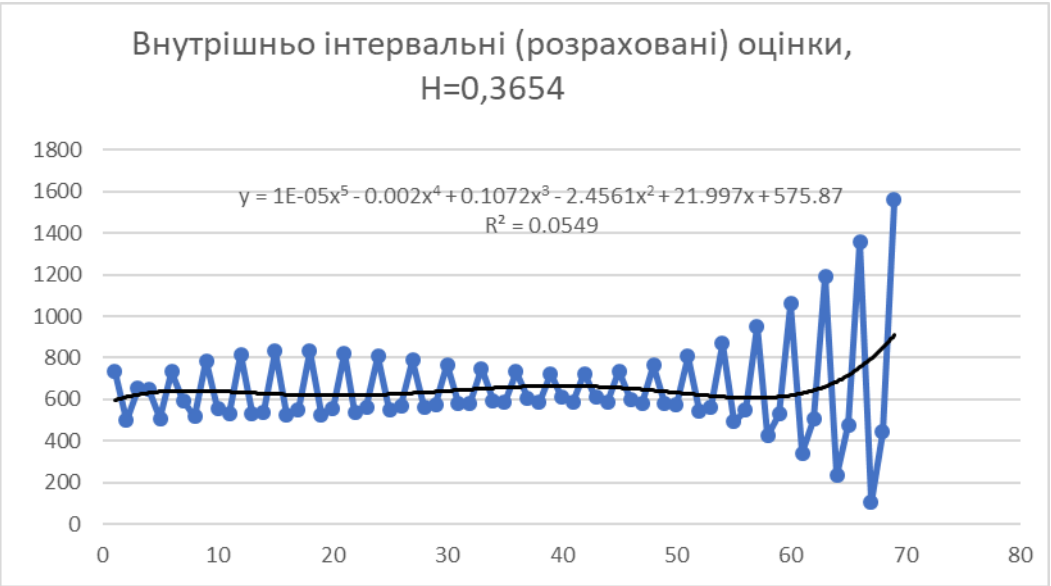


Рисунок 4.9. Внутрішньо інтервальні рівні, побудовані на основі моделей MA3_1 – MA3_3 з перекриттям інтервалів агрегування

На рисунках рис. 4.8 та рис. 4.9 наведено результати застосування процедур інтерполяції внутрішньо інтервальних рівні моделей АЧР.

Висновки до розділу 4

У розділі досліджено метод моделювання та прогнозування антиперсистентних часових послідовностей, а також програмне забезпечення із його реалізації. При цьому представлений опис використаного програмно-апаратного середовища, дана загальна характеристика методу моделювання часових послідовностей часових рядів, подано варіанти схем та методу агрегування АЧР для визначення можливих класів, описані метод побудови інтерполяції часових послідовностей та ланцюгові алгоритмищо застосовані для рішення систем рівнянь, представлені результати проведення чисельних експериментів щодо класифікації антиперсистентних процесів.

Отримані результати свідчать про коректність запропонованих розробок математичних моделей, алгоритмів і програмних засобів, що дозволило встановити ефективну схему класифікації АЧР, з урахуванням особливості структури процесів моделювання антиперсистентних часових послідовностей. При порівнянні результатів застосування алгоритмів класифікації АЧР було встановлено, що найбільше детальною та сталою являється класифікація на основі схем агрегування без перетину елементів,

Розроблений програмний комплекс для моделювання та прогнозування антиперсистентних часових послідовностей здатний обробляти АЧР різних процесів. Він забезпечує функції класифікації, прогнозування та інтерполяції моделей АЧР при умовах, коли вхідна послідовність має малу кількість спостережень та значні варіації досліджуваних процесів.

5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

Для того, щоб кожна людина мала змогу працювати на належному рівні та в належних умовах, необхідно щоб робоче середовище відповідало певним нормам безпеки, а також щоб власне робітники дотримувались певних правил роботи. На сьогоднішній день це питання регулюється групою нормативно-правових актів з охорони праці (далі скорочено НПАОП) та державними санітарними правилами і нормами роботи (далі скорочено ДСанПіН).

Так як розробка програмного забезпечення, насамперед, пов'язана із роботою із комп'ютерною технікою необхідно розглянути відповідні ДСанПіН та НПАОП, які власне стосуються роботи із відповідною технікою в належних для цього приміщеннях

5.1 При створенні певного програмного забезпечення виникає безпосередня необхідність роботи із ЕОМ. При роботі з ЕОМ необхідно, щоб робоче місце відповідало наступному переліку вимог до НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроям»:

- у приміщеннях з джерелами шкідливих виробничих факторів робочі місця операторів мають розміщуватися в ізольованих кабінах, які обладнані повітрообміном;
- будівлі та приміщення, де розміщені робочі місця операторів, мають бути не нижче II ступеня вогнестійкості згідно з ДБН В.1.1.7-2016 «Пожежна безпека об'єктів будівництва. Загальні вимоги»;
- неприпустимим є розташування приміщень категорій А і Б, а також виробництв з мокрими технологічними процесами поряд з приміщеннями, де розташовуються ЕОМ з відеодисплейними терміналами і периферійними пристроями, а також над ними чи під ними;

- у приміщеннях з джерелами шкідливих виробничих факторів робочі місця операторів мають розміщуватися в ізольованих кабінах, які обладнані повітрообміном;
- згідно з ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» не дозволяється розташування приміщень з робочими місцями операторів у підвалах і цокольних поверхах;
- заземлені конструкції, що знаходяться в приміщеннях, де розміщені робочі місця операторів (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном), мають бути надійно захищені діелектричними щитками або сітками з метою недопущення потрапляння працівника під напругу;
- приміщення, де розміщені робочі місця операторів, крім приміщень, у яких розміщені робочі місця операторів великих ЕОМ загального призначення (сервер), мають бути оснащені системою автоматичної пожежної сигналізації.

5.1.1 Електробезпека в приміщенні з ЕОМ

Для того, щоб зменшити кількість виникнення нещасних випадків уражених електричним струмом потрібно провести основні заходи:

- застосування малих напруг і електричний поділ мереж;
- потрібно завжди пам'ятати, що небезпечного електроструму не існує, навіть при ураженні 12В, якщо напруга діє на безпосередньо важливі нервові вузли, керування якими в організмі людини здійснюється електричними сигналами дуже малих напруг.
- контроль ізоляції;
- при порушенні ізоляційного проводу може призвести до летальних випадків, тому контроль ізоляції є необхідним заходом, що попереджує небезпеку ураження електричним струмом.
- захисне заземлення, занулення;

– усі електроприлади які знаходяться в приміщенні згідно з НПАОП 40.1-1.21-98 «Правила безпечної експлуатації електроустановок споживачів» повинні бути заземлені за допомогою нульового захисного провідника. Використання нульового робочого як нульового захисного провідника забороняється. Усі електроприймачі які використовуються в приміщенні, повинні підключатися до електромережі тільки за допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення.

При здійсненні трудової діяльності в приміщенні виникає також електростатичне поле, воно виникає за рахунок тертя: килими, лінолеуми, лаковані покриття, одяг із синтетичної тканини, взуття. Також джерелом електростатичного поля є відео-термінали (ВДТ). Напруженість електростатичного поля залежно від типу ВДТ коливається від 8 до 75 кВ/м відповідно до ГОСТ 12.1.045-84 «Электростатические поля. Допустимые уровни на рабочих местах и требования к проведению контроля».

Для досягнення зниження небезпечних проявів електрики здійснюються такі заходи:

- застосуванням електропровідності поверхні шляхом підтримки у приміщенні відносної вологості на рівні 40-60 %;
- застосуванням іонізації повітря нейтралізаторами статистичної електрики;
- застосуванням у приміщенні з ВДТ підлоги з антистатистичним лінолеумом і проведенням вологого прибирання;
- періодично при вимкненому комп'ютері протирати ледве змоченою мильним розчином бавовняною ганчіркою пил з поверхонь апаратури.

5.1.2 Загальні умови санітарії

Вимоги до приміщення відповідно до ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги»:

- площа не менше 6,0 м²;

- обсяг не менше 20,0 м³;
- робочі місця повинні бути розташовані на відстані не менше ніж 1 м. від стіни з вікном;
- відстань між бічними поверхнями комп'ютерів має бути не меншою за 1,2 м;
- відстань між тильною поверхнею одного комп'ютера та екраном іншого не повинна бути меншою 2,5 м;
- прохід між рядами робочих місць має бути не меншим 1 м.

Конструкція робочого місця оператора має забезпечувати підтримання оптимальної робочої пози з такими ергономічними характеристиками до ДСТУ 8604:2015:

- ступні ніг на підлозі або на підставці для ніг;
- стегна в горизонтальній площині;
- передпліччя вертикально;
- лікті під кутом 70-90 град. до вертикальної площини;
- зап'ястя зігнуті під кутом не більше 20 град. відносно горизонтальної площини, нахил голови 15-20 град. відносно вертикальної площини.

Монітор та клавіатура мають розташовуватися на оптимальній відстані від очей користувача, але не ближче 600 мм, з урахуванням розміру алфавітно-цифрових знаків та символів.

Штучне освітлення приміщення має бути обладнане системою загального рівномірного освітлення. У приміщеннях, де переважають роботи з документами, допускається вживати систему комбінованого освітлення (додатково до загального освітлення встановлюються світильники місцевого освітлення). Загальне освітлення має бути виконане у вигляді суцільних або переривчатих ліній світильників, що розміщуються збоку від робочих місць (переважно зліва) паралельно лінії зору працівників.

5.2 Шкідливі виробничі фактори на підприємстві

Трудова діяльність користувачів комп'ютерів відбувається у певному

виробничому середовищі, яке впливає на їх функціональний стан. Найбільш значимі – фізичні фактори виробничого середовища: електромагнітні хвилі різних частотних діапазонів, електростатичні поля, шум, параметри мікроклімату та ціла низка світлотехнічних показників. Вплив хімічних та біологічних факторів виробничого середовища на користувачів комп'ютера значно менший.

Трудовий процес суттєво впливає на психофізіологічні можливості користувачів комп'ютерів, оскільки їх діяльність характеризується значними статичними фізичними навантаженнями; недостатньою руховою активністю (зменшується швидкість кровообігу, і як результат – зменшується доступ кисню в мозок, що викликає розумове перенапруження); напруженнями сенсорного апарату, вищих нервових центрів, які забезпечують функції уваги, мислення, регуляції рухів. Окрім того, трудовий процес користувачів комп'ютерів відзначається значними інформаційними навантаженнями.

Відповідно до ДСанПН 3.3.2.007-98, працівники, задіяні на роботах, пов'язаних з періодичною або постійною роботою за комп'ютером, піддаються впливу факторів виробничої небезпеки, які поділяються на фізичні та психофізіологічні.

Фізичні фактори виробничої небезпеки:

- підвищений рівень напруги в електричному ланцюзі, замикання якої може пройти через тіло працюючого;
- підвищений рівень рентгенівського випромінювання;
- підвищений рівень ультрафіолетового випромінювання;
- підвищений рівень інфрачервоного випромінювання;
- можливість ураження статичною електрикою;
- запиленість повітря робочого приміщення;
- підвищений вміст важких аероіонів;
- нерівномірний розподіл яскравості в полі зору;
- підвищений рівень пульсації світлового потоку.

Психофізіологічні фактори виробничої небезпеки:

- напруга зору;
- напруга пам'яті;
- напруга уваги;
- тривале статичне напруження;
- відносно великий обсяг інформації, що обробляється в одиницю часу;
- монотонність праці в окремих випадках;
- нераціональна організація робочого місця.

5.2.1 Мікроклімат на робочому місці

Для забезпечення належних умов роботи певного працівника необхідно щоб дотримувались певні норми мікроклімату.

Мікроклімат – це клімат внутрішнього середовища приміщення, який визначається наступними показниками:

- температура повітря;
- відносна вологість повітря;
- швидкість руху повітря;
- інтенсивність теплового (інфрачервоного) опромінення;
- температура поверхні.

Наведені вище показники повинні відповідати до ДСН 3.3.6.042-99 «Державні санітарні норми мікроклімату виробничих приміщень». Так, як робота програміста відноситься до легких робіт категорії Іа, Іб, наведемо відповідні показники мікроклімату згідно визначеної категорії.

- Температура повітря в холодний період року може коливатись в межах від 21 до 24 градусів за Цельсієм, в теплий період року – 22 – 25 градусів за Цельсієм.
- Відносна вологість повітря має коливатись в межах від 40 до 60 % в теплий та холодний період року.
- Швидкість руху повітря має бути від 0.1 до 0.2 м/сек в теплий та холодний період.

- Інтенсивність теплового опромінення не повинна перевищувати 35.0 Вт/м^2 при опроміненні 50% тіла. 70 Вт/м^2 – 25% тіла

5.2.2 Освітлення робочого місця

Така як зорова робота при роботі з екранними пристроями, зокрема текстом, відноситься до категорії середньої точності при якому контраст об'єктів та фон може бути встановленим користувачем, маємо, що за ДБН В.1.1-7:2016 «Пожежна безпека об'єктів будівництва. Загальні вимоги» [7]:

- штучне освітлення при системі комбінованого освітлення всього може коливатись $750 - 400 \text{ лк}$, в тому числі від загального – 200 лк ;
- при системі загального освітлення – від 200 до 300 лк ;
- сукупність нормованих величин показника засліпленості і коефіцієнта пульсації: 40 та 10% відповідно;
- коефіцієнт природної освітленості (далі КПО) природного освітлення має бути в межах $1.5 - 4$, а для суміщеного – $0.9 - 2.4$.

5.3 Дії працівників в надзвичайних ситуаціях

5.3.1 Пожежна безпека

З точки зору пожежонебезпечності, приміщення, в яких проводяться дослідження часової ефективності та роботи із комп'ютером, можна віднести до малопожежонебезпечних, адже в приміщенні присутня документація, електрика – ті елементи, які можуть горіти без вибуху.

Також задля запобігання пожеж необхідно дотримуватись правил пожежної безпеки встановленими правилами пожежної безпеки в Україні [8]:

З боку пожежної безпеки яка стосується приміщень, споруд, тощо, необхідно, щоб:

- на робочих територіях не створювались звалища з горючими речовинами;

- усі під'їзди та проходи до пожежного спорядження, пожежних драбин, евакуаційних виходів мають бути вільними, в належному стані та із освітленням;
- заборонено залишати авто чи інші об'єкти, які заважають наскрізним проїздам до будівлі;
- для куріння мають бути створені окремі ділянки;
- для приміщень мають бути створені та розміщені плани чи схеми евакуації на випадок пожежі.

Для засобів протипожежного захисту (далі СПЗ) мають бути виконані такі вимоги:

- усі СПЗ мають бути справними, та бути в готовому до виконанні роботи стані;
- пожежні гідранти та резервуари повинні бути справними і утримуватись таким чином, щоб пожежні автомобілі могли безперешкодно виконувати забір води;
- пожежні резервуари мають бути захищеними від замерзання води.

Порядок дій у разі пожежі

У разі виявлення ознак пожежі необхідно дотримуватись наступних правил:

- повідомити за телефоном 101 про пожежу;
- припинити роботу в приміщенні;
- якщо є можливість вжити заходів ліквідації вогню за допомогою вогнегасника;
- вжити заходи евакуації;
- якщо є керівник – повідомити його про пожежу;
- в разі необхідності вимкнути електроприлади та відключити електропостачання за винятком протипожежного захисту.

5.3.2 Загальні заходи безпеки в надзвичайних ситуаціях

Відповідно до Кодексу цивільного захисту України підготовка персоналу на підприємствах незалежно від форм власності до дій у надзвичайних ситуаціях здійснюється за спеціально розробленою схемою заходів захисту населення та територій.

Для великих і малих підприємств система заходів захисту від надзвичайних ситуацій включає:

- планування та здійснення необхідних заходів для захисту своїх працівників, об'єктів господарювання;
- розроблення планів локалізації та ліквідації аварій з подальшим погодженням з Державною службою України з надзвичайних ситуацій;
- підтримання у готовності до застосування сил і засобів із запобігання виникненню та ліквідації наслідків надзвичайних ситуацій;
- створення та підтримання матеріальних резервів для попередження та ліквідації надзвичайних ситуацій;
- забезпечення своєчасного оповіщення своїх працівників про загрозу виникнення або при виникненні надзвичайної ситуації.

Нижче наведено приклад основних пунктів необхідних для виконання працівником у разі виникнення надзвичайної ситуації:

- у разі виникнення надзвичайної ситуації працівники повинні діяти тверезо та спокійно та слідувати вказівкам керівництва підприємства, осіб, відповідальних за цивільний захист, протипожежну безпеку, охорону праці, а також представників аварійно-рятувальних, та газорятувальних, пожежних, медичних підрозділів;
- необхідно припинити роботу;
- сповістити про аварію керівника та відповідальну особу;
- якщо можливо – приступити до ліквідації чи локалізації аварії можливим засобами аварії наявними засобами;

- у разі появи незвичного звуку, запаху паленого, відключення комп'ютера та оргтехніки, необхідно негайно припинити роботу і поставити до відома керівника;
- викликати інші аварійно-рятувальні служби;
- у разі виникнення аварійної ситуації через електропостачання необхідно від'єднати електричні прилади від мережі;
- у разі наявності постраждалого необхідно звільнити його від травмуючого фактору, звернутись до медпункту;
- якщо виявлено обрив електропроводів – не варто їх торкатись, а також необхідно сповістити про це безпосереднього керівника.

Висновки до розділу 5

У даному розділі було оглянуто основні питання, які стосуються охорони праці та безпеки в надзвичайних ситуаціях на підприємствах згідно з нормативним актами та санітарними нормами. Зокрема було розглянуто наступні пункти:

- вимоги виконання безпеки на робочому місці;
- шкідливі виробничі фактори на робочому місці;
- дії працівників в надзвичайних ситуаціях.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. НПАОП 0.00-7.15-18. Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. [Текст] / Наказ Міністерства соціальної політики від 14.02.2018 р. № 207, К., 2018.
2. ДБН В.1.1-7:2016. Пожежна безпека об'єктів будівництва. Загальні вимоги [Текст] Наказ Держбуду України від 03.12.2016 року № 88. – К., 2016.
3. ДСанПіН 3.3.2-007-98. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин [Текст] / Постанова Головного державного санітарного лікаря України від 10 грудня 1998 р. № 7 – К., 1998.
4. НПАОП 40.1-1.21-98. Правила безпечної експлуатації електроустановок споживачів[Текст] / Наказ Держнаглядохоронпраці України від 09.01.1998 № 4 – К., 1999.
5. ГОСТ 12.1.045-84. Электростатические поля. Допустимые уровни на рабочих местах и требования к проведению контроля[Текст] / Постановление Госстандарта СССР от 17 сентября 1984 г. N 3236 – М., 1985.
6. ДСН 3.3.6.042-99. Державні санітарні норми мікроклімату виробничих приміщень [Текст]/ Постанова Головного Державного санітарного лікаря України від 01.12.1999 № 42 - К., 1999.
7. ДБН В.1.1-7:2016. Пожежна безпека об'єктів будівництва. Загальні вимоги [Текст] Наказ Держбуду України від 03.12.2016 року № 88. – К., 2016.
8. НАПБ А.01-001-2015. Правила пожежної безпеки в Україні [Текст]. Редакція від 03.10.2017.
9. Кодекс цивільного захисту України [Текст]/ №5403-VI від 02.10.2012 року.
10. Сторінка веб-ресурсу <https://osvita-docs.com/node/41>. Станом на 02.12.2019.

ВИСНОВКИ

У роботі отримано результатів досліджень щодо можливостей та ефективності алгоритмів, призначених для класифікації, моделювання та прогнозування показників антиперсистентних процесів. При цьому були розглянуті структури процедур аналізу та класифікації антиперсистентних процесів, проведено аналіз напрямків щодо удосконалення алгоритмів класифікації та короткострокового прогнозування антиперсистентних часових послідовностей (АЧП), запропонована методика та процедури дослідження АЧР шляхом порівняльного аналізу чисельної ефективності та точності алгоритмів класифікації та інтерполяції даних моделей АЧП.

На основі зазначеної методики класифікації та прогнозування АЧР проведені дослідження функціональної ефективності алгоритмів моделювання. В роботі були визначені та представлені особливості структури процесів моделювання антиперсистентних числових послідовностей. При порівнянні алгоритмів класифікації АЧП було встановлено, що найбільше детальною та сталою являється класифікація на основі схем агрегування даних АЧП без перетину у послідовностях елементів що узагальнюються, тобто рівнів часових рядів. Отримані результати були використані при формування програмного забезпечення для дослідження моделей дискретних антиперсистентних процесів.

Практичне значення мають результати щодо засобів комп'ютерного моделювання АЧП, розробки та порівняльного аналізу алгоритмів моделювання, результати порівняльного аналізу чисельної ефективності алгоритмів класифікації та процедури моделювання антиперсистентних числових послідовностей.

Реалізовано програмний комплекс, здатний моделювати агреговані моделі антиперсистентних ЧР, виконувати інтерполяцію внутрішньо інтервальних значень, отриманих при прогнозуванні за певним інтервалом узагальнення, будувати систему моделей для різних схем агрегування даних при моделюванні антиперсистентних процесів.

Для розробки програмного комплексу використовувались технології VCL. Завдяки чому проектування та розробка інтерфейсу була легкою та швидкою, також це дозволить з легкістю розширювати функціонал програмного комплексу у разі необхідності, без втручання в існуючі модулі.

Програмний комплекс було про тестовано та відлагоджено. Для тестування функції розрахунку моделі використано методи «чорної скриньки» та «білої скриньки».

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Прогнозування і моделювання (Oracle Crystal Ball) [Електронний ресурс] – Режим доступу: http://ubc-corp.ru/ru/oracle_crystal_ball.
2. Прогнозування в Predictive Planning (Oracle Crystal Ball) [Електронний ресурс] – Режим доступу: <https://ivan-shamaev.ru/predictive-planning-oracle-crystal-ball/>
3. Tahseen A., Aqil S., Burney Cemal A. A New Quantile Based Fuzzy Time Series Forecasting Model [Електронний ресурс] – Режим доступу: <https://publications.waset.org/14214/pdf>
4. Qiang S., Brad S., Forecasting enrollments with fuzzy time series - Part I [Електронний ресурс] – Режим доступу: <https://www.sciencedirect.com/science/article/pii/S016501149390355L>
5. Sheng T., Yi-Chung C. Deterministic fuzzy time series model for forecasting enrollments [Електронний ресурс] – Режим доступу: <https://www.sciencedirect.com/science/article/pii/S0898122107001630>
6. Tahseen A., Jilani S., Aqil B., Ardil C. Multivariate High Order Fuzzy Time Series Forecasting for Car Road Accidents [Електронний ресурс] – Режим доступу: https://www.researchgate.net/publication/285870449_Multivariate_High_Order_Fuzzy_Time_Series_Forecasting_for_Car_Road_Accidents
7. Скалозуб В.В., Клименко И.В. Оценка и прогнозирование параметров временных рядов вагонопотоков на основе специализированной процедуры фрактального анализа. *Наук. вісн. Херсон. держ. ун-ту. Сер.: Економічні науки.* (Google Scholar, Index Copernicus). 2016. Вип. 20, ч. 2. С. 189 – 192. – (0,44 друк. арк., особисто автору – 0,21 друк. арк., запропонована процедура класифікації антиперсистентних часових рядів).
8. Скалозуб В.В., Клименко І.В. Развитие процедур анализа та прогнозування недетермінованих технолого-економічних процесів на основі показників хаотичної динаміки. *Економіка: реалії часу.* (Google

- Scholar, Index Copernicus*). 2016. № 4 (26). С. 149 – 154. – (0,39 друк. арк., особисто автору – 0,11 друк. арк., класифікації антиперсистентних часових рядів).
9. Скалозуб В.В., Клименко И.В. Метод планирования недетерминированных процессов эксплуатации парка железнодорожных технических систем. *Наука та прогрес транспорту. Вісник Дніпропетровського національного університету залізничного транспорту. Сер.: Економічні науки*. (Google Scholar, Index Copernicus). 2018. № 5 (77). С. 7 – 18. – (1,01 друк. арк., особисто автору – 0,61 друк. арк., формування та застосування індивідуальних моделей груп парків технічних систем).
 10. Клименко И.В. Применение процедуры агрегирования для анализа и оперативного прогнозирования недетерминированных временных рядов технологических процессов железнодорожного транспорта. *Проблеми економіки транспорту* : тези доп. XIII Міжнародної науково-практичної конференції, м. Дніпропетровськ, 23-24 квітня 2015р. Дніпропетровськ 2015. С. 167-168. (0,11 друк. арк.).
 11. Б. Безручко и Д. Смирнов, Математическое моделирование и хаотические временные ряды, Саратов: Гос. УНЦ "Колледж", 2005, р. 320с..
 12. Л. Сергеева, Моделирование поведения экономических систем методами нелинейной динамики (теории хаоса), Запорожье: ЗГУ, 2002, р. 227.
 13. Q. Song, and B. S. Chissom, "Forecasting enrollments with fuzzy time series — Part I," *Fuzzy Sets and Systems*, vol. 54, issue 1, 1993a, pp. 1–9.
 14. B. S. Chissom, "Fuzzy time series and its models," *Fuzzy Sets and Systems*, vol. 54, issue 3, 1993b, pp. 269-277.
 15. Q. Song, and B. S. Chissom, "Forecasting enrollments with fuzzy time series — Part II," *Fuzzy Sets and Systems*, vol. 62, 1994, pp. 1-8.
 16. S. M. Chen, "Forecasting enrollments based on fuzzy time series," *Fuzzy Sets and Systems*, vol. 81, 1996, pp. 311-319.

- 17.Q. Song, "A note on fuzzy time series model selection with sample autocorrelation functions," *Cybernetics and Systems: An International Journal*, vol. 34, 2003, pp. 93-107.
- 18.Q. Song, and R.P. Leland, "Adaptive learning defuzzification techniques and applications," *Fuzzy Sets and Systems*, vol. 81, 1996, pp. 321-329.
- 19.J. R. Hwang, S. M. Chen, and C. H. Lee, "Handling forecasting problems using fuzzy time series", *Fuzzy Sets and Systems*, vol. 100, 1998, pp.217-228.
- 20.K. Huarng, "Heuristic models of fuzzy time series for forecasting," *Fuzzy Sets and Systems*, vol. 123, issue 3, 2001a, pp. 369-386
- 21.K. Huarng, "Effective lengths of intervals to improve forecasting in fuzzy time series," *Fuzzy Sets and Systems*, vol. 123, issue 3, 2001b, pp. 387-394.
- 22.S.-M. Chen, "Forecasting enrollments based on high-order fuzzy time series," *Cybernetics and Systems: An International Journal*, vol. 33, pp. 1-16.
- 23.S.-M. Chen, and J.-R. Hwang, "Temperature prediction using fuzzy time series, *IEEE Transactions on Systems, Man, and Cybernetics — Part B*," *Cybernetics*, vol. 30, 2000, pp. 263-275.
- 24.R. -C. Tsaur, J. -C. O. Yang, and H. -F. Wang, "Fuzzy relation analysis in fuzzy time series model," *Computers and Mathematics with Applications*, vol. 49, 2005, pp. 539-548.
- 25.H. Li, and R. Kozma, "A dynamic neural network method for time series prediction using the KIII model," In proceedings of the 2003 International Joint Conference on Neural Networks, vol. 1, 2003, pp. 347-352.
- 26.S. F. Su, and S. H. Li, "Neural network based fusion of global and local information in predicting time series," In proceedings of the 2003 IEEE International Joint Conference on Systems, Man and Cybernetics, vol. 5, 2003 pp. 4445-4450.
- 27.J. Sullivan, and W. H. Woodall, "A comparison of fuzzy forecasting and Markov modeling," *Fuzzy Sets and Systems*, vol. 64, 1996, pp. 279-293.

- 28.L. -W. Lee, L. -W. Wang, S. -M. Chen, and Y.-H. Leu, "Handling forecasting problems based on two-factor high-order time series," IEEE Transactions on Fuzzy Systems, vol. 14, issue 3, 2006, pp. 468-477.
 - 29.T. A. Jilani, S. M. A. Burney, and C. Ardil, "Multivariate high order fuzzy time series forecasting for car road accidents," International Journal of Computational Intelligence, vol. 4, issue 1, 2007b, pp. 15-20.
 - 30.T. A. Jilani, and S. M. A. Burney, "M-factor high order fuzzy time series forecasting for road accident data," In IEEE-IFSA 2007," World Congress, Cancun, Mexico, June 18-21, 2007. Forthcoming in Book series Advances in Soft Computing, Springer-Verlag, 2007a.
 - 31.S. -T. Li, and Y. -C. Cheng, "Deterministic fuzzy time series model for forecasting enrollments," Computers and Mathematics with Applications, vol. 53, 2007, pp. 1904-1920.
 - 32.L. A. Zadeh, "Fuzzy sets," Information and Control, vol. 8, 1996, pp. 338-353.
 - 33.R. Koenker, "Quantile Regression", Cambridge University Press, NY- 2005.
- охорона праці
- 34.НПАОП 0.00-6.03-93. Порядок опрацювання та затвердження власником нормативних актів про охорону праці, що діють на підприємстві.
 - 35.НПАОП 0.00-4.15-98. Положення про розробку інструкцій з охорони праці
 - 36.НПАОП 0.00-4.12-05. Типове положення про навчання з питань охорони праці.
 - 37.ДСН 3.3.6.039-99. Державні санітарні норми виробничої загальної та локальної вібрації.
 - 38.ДСанПІН 3.3.2.007-98. Робота з візуальними дисплейними терміналами електронно-обчислювальних машин.
 - 39.ДСН 3.3.6.042-99. Державні санітарні норми мікроклімату виробничих приміщень.
 - 40.НПАОП 0.00-7.15-18. Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями.

41. oppb.com [Електронний ресурс] – Електронні дані – Режим доступу: <http://oppb.com.ua> (дата звернення 12.07.2019 р.) – Забезпечення пожежної безпеки.
42. lubbook [Електронний ресурс] – Електронні дані – Режим доступу: <https://lubbook.org> (дата звернення 05.03.2018 р.) – Мікроклімат приміщень для роботи з комп'ютерною технікою.
43. Q. Song, B.S. Chissom, Forecasting enrollments with fuzzy time series—part I, *Fuzzy Sets and Systems* 54 (1993) 1–9.
44. Q. Song, B.S. Chissom, Fuzzy time series and its models, *Fuzzy Sets and Systems* 54 (1993) 269–277.
45. Q. Song, B.S. Chissom, Forecasting enrollments with fuzzy time series—part II, *Fuzzy Sets and Systems* 62 (1994) 1–8.
46. J. Sullivan, W.H. Woodall, A comparison of fuzzy forecasting and Markov modeling, *Fuzzy Sets and Systems* 64 (1994) 279–293.
47. S.-M. Chen, Forecasting enrollments based on fuzzy time series, *Fuzzy Sets and Systems* 81 (1996) 311–319.
48. S.-M. Chen, Forecasting enrollments based on high-order fuzzy time series, *Cybernetics and Systems: An International Journal* 33 (2002) 1–16.
49. S.-M. Chen, J.-R. Hwang, Temperature prediction using fuzzy time series, *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics* 30 (2000) 263–275.
50. J.R. Hwang, S.M. Chen, C.H. Lee, Handling forecasting problems using fuzzy time series, *Fuzzy Sets and System* 100 (1998) 217–228.
51. K. Huarng, Heuristic models of fuzzy time series for forecasting, *Fuzzy Sets and Systems* 123 (2001) 369–386.
52. R.-C. Tsaur, J.-C. Yang, H.-F. Wang, Fuzzy relation analysis in fuzzy time series model, *Computers & Mathematics with Applications* 49 (2005) 539–548.
53. T.-S. Lee, C.-C. Chiu, F.-C. Lin, Prediction of the unemployment rate using fuzzy time series with Box–Jenkins methodology, *International Journal of Fuzzy Systems* 3 (2001) 577–585.

- 54.C.-C. Tsai, S.-J. Wu, A study for second-order modeling of fuzzy time series, Proceedings of the IEEE International Conference on Fuzzy Systems (1999) 719–725.
- 55.K. Huarng, Effective lengths of intervals to improve forecasting in fuzzy time series, Fuzzy Sets and Systems 123 (2001) 387–394.
- 56.S.-T. Li, Y.-P. Chen, Natural partition-based forecasting model for fuzzy time series, in: IEEE International Conference on Fuzzy Systems, Budapest, Hungary, 2004, pp. 25–29.
- 57.S.-T. Li, Y.-C. Cheng, A deterministic forecasting model for fuzzy time series, in: The Fourth IASTED International Conference on Computational Intelligence, Calgary, Alberta, Canada, 2005, pp. 25–30.
- 58.G. J. Klir and B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice Hall, India, 2005, Ch. 4.
- 59.H. Ishibuchi, R. Fujioka and H. Tanaka, "Neural Networks that Learn from Fuzzy If-Then Rules", IEEE Transactions on Fuzzy Systems, Vol. 1, No. 1, pp.85-97, 1993.
- 60.H. J. Zimmerman, Fuzzy set theory and its applications, Kluwer Publishers, Boston, MA, 2001.
- 61.K. Huarng, "Heuristic models of fuzzy time series for forecasting," Fuzzy Sets Systems, vol. 123, no. 3, pp. 369–386, 2001a.
- 62.K. Huarng, "Effective Lengths of Intervals to Improve Forecasting in Fuzzy Time Series," Fuzzy Sets System, Vol. 123, No. 3, pp. 387–394, 2001b.
- 63.L. W. Lee, L. W. Wang, S. M. Chen, "Handling Forecasting Problems Based on Two-Factors High-Order Time Series," IEEE Transactions on Fuzzy Systems, Vol. 14, No. 3, pp.468-477, Jun. 2006.
- 64.Melike Sah and Y. D. Konstsntin, "Forecasting Enrollment Model based on first-order fuzzy time series," Published in proc., International Conference on Computational Intelligence, Istanbul, Turkey, 2004.
- 65.Q. Song and B. S. Chissom, "Forecasting Enrollments with Fuzzy Time Series—Part I," Fuzzy Sets and System, Vol. 54, No. 1, pp. 1–9, 1993 a.

- 66.R. R. Yager and P. P. D. Filev, Essentials of FUZZY MODELING and Control, John Wiley and Sons, Inc. 2002.
- 67.S. M. Chen, "Forecasting Enrollments Based on High-Order Fuzzy Time Series," Cybernetic Systems, Vol. 33, No. 1, pp. 1–16, 2002.
- 68.S. Park and T. Han, "Iterative Inversion of Fuzzified Neural Networks," IEEE Transactions on Fuzzy Systems, Vol. 8, No. 3, pp. 266- 280, 2000
- 69.S.P. Chatman, Short-term forecasts of the number and scholastic ability of enrolling freshman by academic divisions, Res. Higher Educ. 25(1) (1986) 68-81.
- 70.A. Freeman and D.M. Skapura, Neural Networks: Algorithms, Applications and Programming Techniques (Addison-Wesley, Reading, MA, 1991).
- 71.D.E. Gardner, Weight factor selection in double exponential smoothing enrollment forecasts, Res. Higher Educ. 14(1) (1981) 49-56.
- 72.S.A. Hoenack and W.C. Weiler, The demand for higher education and institutional enrollment forecasting, Economic Inquiry 17 (1979) 89-113.
- 73.B. Kosko, Neural Networks and Fuzzy Systems (Prentice Hall, Englewood Cliffs, NJ, 1992).
- 74.C.T. Lin and C.S. Lee, Neural-network-based fuzzy logic control and decision system, IEEE Trans, on Computers 40(12) (1991) 1320-1336.
- 75.Y. Miyata, A user's guide to PlaNet version 5.6, 1991.
- 76.M.B. Paulsen, A practical model for forecasting new freshmen enrollment during the application period, College and University 64(4) (1989) 379-391.
- 77.A. Адаменко, «Динамічні нечіткі множини,» *Збірник наукових праць Харківського університету Повітряних сил*, т. №1, pp. 90-93, 2010.
78. О. Мулеса, В. Снийюк та С. Герзанич, «Метод нечіткої класифікації на основі послідовного аналізу вальда,» *Automation of technological and business processes*, т. №11, pp. 35-42, 2020.
79. В. Иванов, Анализ временных рядов и прогнозирование экономических показателей, Харьков: ХНУ, 1999, p. 230

80. «Расчет показателя Херста с целью выявления трендовости (персистентности) финансовых рынков [Электронный ресурс],» 2010
81. Н. Б. Паклин и В. И. Орешков, Бизнес-аналитика: от данных к знаниям, СПб.: Питер, 2009, р. 624.
82. Э. Петерс, Фрактальный анализ финансовых рынков. Применение теории хаоса в инвестициях и экономике, М.: Интернет-трейдинг, 2004, р. 304.
83. В. Кравець та Т. О. Гапоненко, «Фрактальний аналіз валютного ринку за допомогою моніторингу показника Херста,» *Бізнес Інформ*, т. №11, рр. 125-131, 2015.
84. С. Кузнецов и О. Гладковский, «Фрактальный анализ котировок ВТБ,» Сибирская Академия Государственной службы, Новосибирск, 2010.

ДОДАТКИ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАТВЕРДЖУЮ

Перший проректор Українського
державного університету
науки і технологій

Анатолій Радкевич

ДОСЛІДЖЕННЯ МОДЕЛЕЙ АНТИПЕРСИСТЕНТНИХ ЧАСОВИХ
ПОСЛІДОВНОСТЕЙ

РОБОЧИЙ ПРОЕКТ

ЛИСТ ЗАТВЕРДЖЕННЯ

1116130.01222-01-ЛЗ

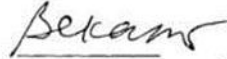
Завідувач кафедри КІТ

Вадим ГОРЯЧКІН



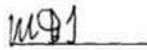
Керівник розробки

Владислав СКАЛОЗУБ



Виконавець

Данило ШАПОВАЛ



Нормоконтролер

Олена КУРОП'ЯТНИК



2021

ЗАТВЕРДЖЕНО
1116130.01165-01-ЛЗ

Дослідження моделей антиперсистентних часових послідовностей
Технічне завдання

1116130.01165-01

Аркушів 22

1116130.01165-01

АНОТАЦІЯ

Документ 1116130.01165-01 «Дослідження моделей антиперсистентних часових послідовностей» Технічне завдання, що входить до складу програмної документації до дипломного проекту.

У даному документі представлене призначення та область застосування програми, основні вимоги, стадії та строки виконання проекту, технічні та техніко-економічні показники, що пред'являються до програми.

3
1116130.01165-01
ЗМІСТ

ВСТУП	5
1 ПІДСТАВА ДЛЯ РОЗРОБКИ	8
2 ПРИЗНАЧЕННЯ РОЗРОБКИ	9
2.1 Функціональне призначення	9
2.2 Експлуатаційне призначення	9
3 ВИМОГИ ДО ПРОГРАМИ	10
3.1 Вимоги до функціональних характеристик	10
3.1.1 Вхідні дані	10
3.1.2 Вихідні дані	10
3.2 Вимоги до надійності	11
3.3 Умови експлуатації	11
3.4 Вимоги до складу та параметрів технічних засобів	12
3.5 Вимоги до інформаційної та програмної сумісності	12
3.6 Вимоги до маркування та пакування	13
3.7 Вимоги до транспортування та зберігання	13
4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ	14
5 КОШТОРИС НА РОЗРОБКУ ПЗ	15
5.1 Загальні положення	15
5.2 Розрахунок основної заробітної плати	15
5.3 Розрахунок накладних витрат	16
5.3.1 Розрахунок витрат на електроенергію	17
5.3.2 Розрахунок амортизаційних відрахувань	18
5.3.3 Комунальні послуги	19
5.3.4 Зведений розрахунок накладних витрат	19
5.3.5 Розрахунок витрат на створення програмного продукту	20
6 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ	21
7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙОМУ	22

СПИСОК ЛІТЕРАТУРИ.....	23
------------------------	----

ВСТУП

«Дослідження моделей антиперсистентних часових послідовностей» – це програмний комплекс для аналізу та дослідження властивостей часових рядів (НЧР), створений на основі нової категорії нечітких моделей. Для моделювання використані спеціальні форми обліку часових інтервалів між рівнями часового ряду (ЧР). При сепарабельній формі моделі ЧР нерівномірні (або нечіткі) інтервали виділяються у окрему складову моделі НЧР. Вони є однією окремою складовою вектору характеристик процесів. Ця складова моделюється окремо, коли враховуються лише послідовності величин часових інтервалів. На результати моделювання таких інтервалів «накладаються» результати моделювання інших характеристик у відповідності до порядку (номера) їх розміщення у початкових НЧР. Тобто при моделюванні окремо для кожної властивості застосовуються процедури обробки з рівномірним кроком (моделі FTS, Fuzzy Time Series першого і другого порядку).

Різноманітні технологічні, виробничі, інформаційні, лікувальні та інші процеси у складних системах можуть мати значний ступень невизначеності деяких параметрів і характеристик. При цьому для них на практиці можливо отримання лише нерегулярні у часі послідовності даних, через різні за величинами інтервали. Для аналізу таких процесів є лише нерівномірні у часі послідовності даних, які характеризують зміни різноманітних показників систем. Великі труднощі та значна вартість оперативних та достовірних даних про оцінки параметрів програмних систем на практиці застосовують підходи і моделі нечіткого моделювання та аналізу рядів (НЧР). У якості прикладів такого типу об'єктів розглянуто такі сфери. В проектуванні та розробці програмного забезпечення, контроль показників якості версій деякого програмного забезпечення широкого застосування, а також оцінка ефективності програмних комплексів із розробки ігрових платформ. Для таких об'єктів встановити певні детерміновані послідовності отримання та значень контрольованих параметрів не представляється можливим.

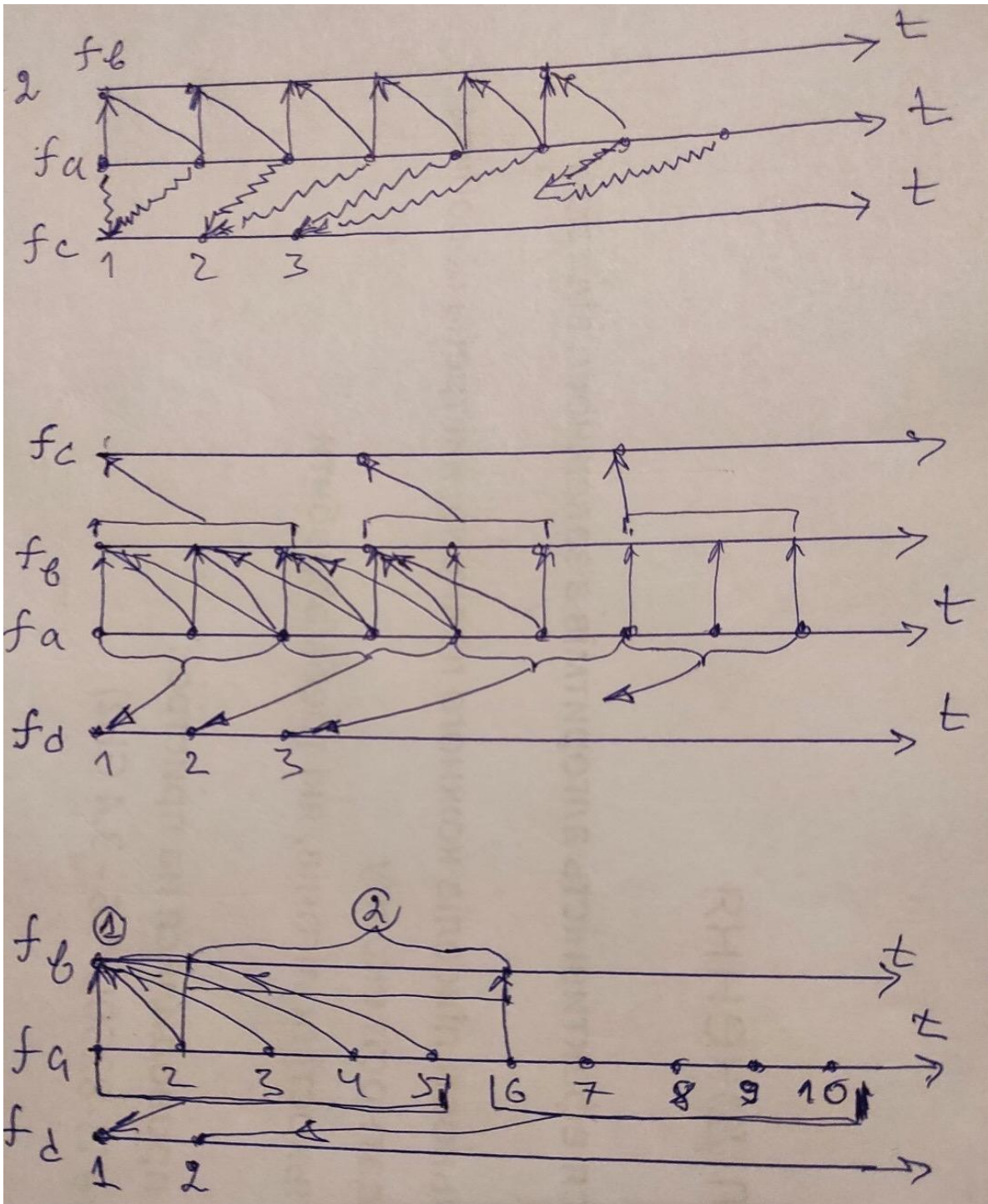


Рисунок Д.1. Порівняльні схеми варіантів агрегування вихідних даних для різних типів класифікації ЧР ($k=2$, $k=3$, $k=5$),

На рис. Д.1 наведено приклади варіантів агрегування даних для побудови узагальнених коректних математичних моделей недетермінованих процесів, сформованих за даними спостережень. Рисунок демонструє широкі можливості утворення агрегованих моделей процесів у вигляді ЧР. При цьому в

дослідженнях необхідно встановити, по-перше, при якій формі агрегування забезпечується можливість класифікації вихідних ЧР, а по-друге, запропонувати алгоритми прогнозування за агрегованими моделями. Такі алгоритми мають дозволяти виконувати інтерполяцію – розраховувати внутрішні рівні показників процесів ЧР, використовуючи оцінки інтегрованих величин, отриманих за моделями трендів.

1 ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є наказ ректора Дніпровського національного університету залізничного транспорту імені академіка В. Лазаряна проф. Пшінька О. М. № 833ст від 12.11.2020 р. «Про призначення керівників та затвердження тем магістерських робіт» факультету «Комп'ютерні технології і системи» за спеціальністю 121 «Інженерія програмного забезпечення» по кафедрі «Комп'ютерні інформаційні технології».

У відповідності з наказом, тема дипломного проекту – «Дослідження моделей антиперсистентних часових послідовностей» Керівник проекту проф. Скалозуб В. В.

2 ПРИЗНАЧЕННЯ РОЗРОБКИ

2.1 Функціональне призначення

Програмний комплекс призначений для аналізування та дослідження математичних властивостей нечітких часових рядів. Функціональним призначенням програми є наступне:

- побудова математичної моделі на основі вдосконаленого методу квантильної моделі;
- побудова сепарабельної моделі відносно вхідного нечіткого часового ряду;
- розмноження нечіткого часового ряду методом бутстрепа [8].

2.2 Експлуатаційне призначення

Експлуатаційне призначення програмного комплексу полягає в автоматизовані обробці нечітких часових рядів різноманітних технологічних, виробничих, інформаційних, лікувальних та інших процесів, швидкому порівнянні отриманих моделей, а також прогнозування. Експлуатаційними параметрами порівняння точності, які визначаються за призначенням програмного комплексу, є:

- порівняння значень нечіткого часового ряду з побудованою моделлю на графіку;
- прогнозування нечіткого часового ряду на основі розробленого алгоритму;

3.1 Вимоги до функціональних характеристик

Вимоги до функціональних характеристик програмного комплексу наступні:

- завантаження нечітких часових рядів з файлу Microsoft Office Excel;
- побудова математичної моделі за вдосконаленим методом квантильної моделі;
- розмноження нечіткого часового ряду методом бутстреп [8];
- побудова варіантів нечітких часових рядів на основі декількох рівнів;
- побудова сепарабельної моделі;
- відображення нечітких часових рядів на графіку;
- відображення математичної моделі на графіку;
- збереження результатів в файл Microsoft Office Excel;

3.1.1 Вхідні дані

Вхідні дані для користувача наступні:

- певний процес, представлений у вигляді нечіткого часового ряду (час представлений у вигляді дати формату дд.мм.рр, значення речового числа)
- завантаження нечіткого часового ряду з файлу Microsoft Office Excel (файл формату .xls/xlsx);
- параметр Карра (ціле число);
- параметр прогнозування (ціле число більше нуля);
- параметр фіксовані точки (ціле число більше нуля);
- параметр розмір нового ряду (ціле число більше або рівне $2 \cdot \text{фіксовані точки}$);
- параметр кількість точок (ціле число більше одного).

3.1.2 Вихідні дані

Вихідні дані для користувача наступні:

- відображення нечіткого часового ряду в табличному вигляді;
- відображення побудованих моделей нечіткого часового ряду в табличному вигляді;
- відображення сепарабельної моделі нечіткого часового ряду в табличному вигляді;
- відображення розмноженого нечіткого часового ряду в табличному вигляді;
- відображення нечіткого часового ряду на графіку;
- відображення побудованих моделей нечіткого часового ряду на графіку;
- відображення сепарабельної моделі нечіткого часового ряду на графіку;
- відображення розмноженого нечіткого часового ряду на графіку;

- збереження результатів у файлі Microsoft Office Excel формату .xls/xlsx;
- виведення підказок користувачу;
- виведення помилок користувача.

3.2 Вимоги до надійності

Одним із критеріїв правильного функціонування програмного комплексу є забезпечення надійності роботи програми.

Програмний комплекс повинен:

- відповідати специфікації;
- відповідати функціональним характеристикам;
- забезпечити контроль вхідних і вихідних даних. У випадку неправильного вводу даних – видати повідомлення про помилку;
- забезпечити надійне збереження даних.

3.3 Умови експлуатації

Для забезпечення надійного функціонування програмного комплексу, користувачеві необхідно дотримуватися таких умов:

- програмний комплекс повинен використовуватись у приміщеннях, які відповідають умовам роботи ЕОМ і мають такі кліматичні, санітарні та гігієнічні умови, які відповідають ДСанПіН 3.3.2-007-98 [1];
- для роботи з програмним комплексом, користувач повинен мати базові навички роботи з Microsoft Office Excel, та ознайомлений с керівництвом користувача;
- програмний комплекс повинен використовуватись у приміщеннях з наступними кліматичними умовами: температура навколишнього повітря – 21– 25 °С, відносна вологість повітря – 40-60 %;

Мінімальна кількість персоналу, необхідного для роботи програми 1 штатна одиниці – 1 користувач програми – користувач.

Користувач програмного комплексу повинен мати практичні навички роботи за ПК.

3.4 Вимоги до складу та параметрів технічних засобів

Для коректного функціонування програми апаратна частина повинна задовольняти наступним умовам:

- процесор AMD Ryzen 2200g; оперативна пам'ять 1 Гб або більше; CD або DVD-привод;
- USB-роз'єм; клавіатура; комп'ютерна «миша».

3.5 Вимоги до інформаційної та програмної сумісності

Програмний комплекс розрахований на операційну систему, що має Microsoft Office Excel.

3.6 Вимоги до маркування та пакування

Упаковка програмного комплексу, включаючи документацію, повинна бути захищена від пошкоджень різного роду (механічних, кліматичних).

На упаковці повинно бути вказано назву програмного комплексу, номер версії, якщо вона змінювалась, мінімальні системні вимоги .

На зворотній стороні упаковки вказується розробник та його юридична адреса. На рис. 3.1 приведений приклад маркування.

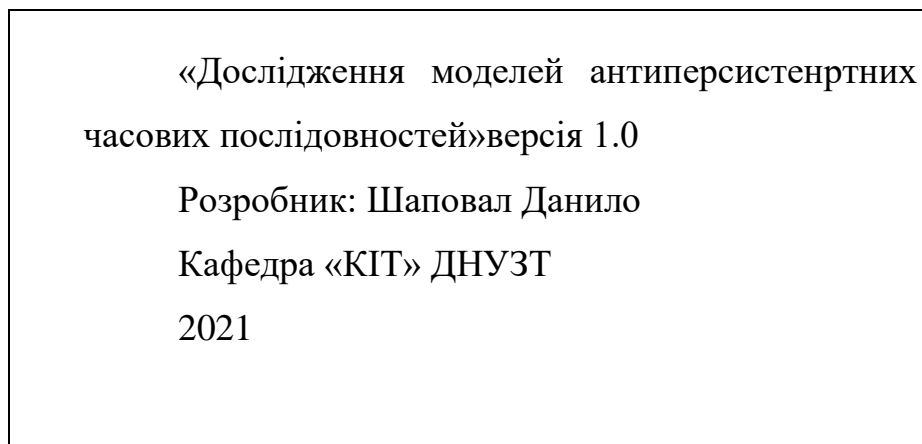


Рисунок 3.1 – Приклад маркування

3.7 Вимоги до транспортування та зберігання

Транспортування повинно проводитися довіреною особою. Воно проводиться в упаковці, яка захищає носії з програмним комплексом від різного роду пошкоджень.

Місце зберігання програмного комплексу повинно бути сухим, з відсутністю пилу та з низьким коефіцієнтом відносної вологості повітря. Строк зберігання програмного комплексу залежить від носія інформації.

Транспортування буде здійснюватися на таких носіях: CD/DVD-RW, флеш, також через глобальну всесвітню систему інформаційного обміну Internet.

4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

До складу програмної документації має входити технічне завдання та робочий проект.

До складу робочого проекту мають входити:

- специфікація;
- текст програми;
- опис програми;
- керівництво користувача. Керівництво з моделювання нерегулярної часової послідовності.

Вся документація до програми повинна задовольняти вимогам державного стандарту до оформлення програмних документів [3].

5 КОШТОРИС НА РОЗРОБКУ ПЗ

5.1 Загальні положення

Техніко-економічне обґрунтування (ТЕО) — це розрахунок економічної доцільності здійснення проекту, заснований на порівняльній оцінці витрат і результатів ефективності використання, а також строку окупності вкладень. ТЕО — це виваженість кожного вашого кроку в реалізації задуманого.

Техніко-економічне обґрунтування (ТЕО) являє собою комплексний передпроектний документ, базується на аналізах і розрахунках різних показників. На підставі всіх розрахунків і аналітичних даних у ТЕО даються висновки про економічну доцільність реалізації проекту, дається оцінка перспективам впровадження проекту.

На першому етапі для розрахунку величин трудових витрат розробників необхідно оцінити розмір програмного забезпечення. Різниця між методиками для оцінки трудовитрат залежить від типу критерію оцінки якості (кількісний або якісний)[3].

Згідно моделі COCOMO, розмір проекту S вимірюється в рядках коду LOC (KLOC), а трудовитрати в людино-місяцях.

$$E = a \cdot S^b \cdot EAF,$$

де E – витрати праці на проект (в людино-місяцях);

S^b – розмір коду (в KLOC);

EAF – фактор уточнення витрат (effort adjustment factor).

Для простих систем, $a = 2,4$; $b = 1,05$.

Припустимо, що розмір програмного коду програмного засобу – 872 рядків:

$$E = 2,4 \cdot 0,872^{1,05} \cdot 1 = 2 \text{ люд./міс.}$$

Отже, згідно моделі COCOMO, орієнтовні трудовитрати на проект складуть приблизно 2,4 людино-місяці.

Нижче наведені розрахунки вартості розробки «Моделювання нечітких часових рядів». Основними статтями витрат прийняті:

- основна заробітна плата;
- відрахування на соціальні потреби;
- накладні витрати;
- витрати на персональний комп'ютер і ліцензійні базові програмні засоби.

5.2 Розрахунок основної заробітної плати

Основна заробітна плата (ОЗП) оцінює працю інженера-програміста зі створення програмного продукту і визначається виходячи з кількості розробників, часу виконання розробки (годин), а також заробітної плати в розрахунку на одну годину [4]. Розрахунок заробітної платні проводиться по формі табл. 5.1.

Таблиця 5.1 – Фонд місячної заробітної плати

№ п/п	Посада виконавця	Оклад, грн/міс	Кількість		Сума зарплати грн
			чоловік	місяців	
1	Інженер-програміст	11000	1	2	22000

Описаний в проекті програмний продукт буде розроблений одним програмістом в період з 21.10.19 до 13.12.19, що складає 40 дні або 8 робочих тижнів. Витрати робочого часу прийняті за 40 годин у тиждень. Погодинна ставка кваліфікованого інженера–програміста складає 55 грн/год. Витрати робочого часу за період участі в проекті ($T_{розр}$) визначаються за формулою:

$$T_{розр} = N_{чол} \cdot N_{тиж} \cdot N_{год},$$

де $N_{чол}$ – кількість виконавців, чол.;

$N_{тиж}$ – тривалість розробки в тижнях;

$N_{год}$ – витрати робочого часу за тиждень, год;

$$T_{розр} = 1 \cdot 8 \cdot 40 = 320 \text{ чол./год.}$$

ОЗП визначається за формулою:

$$ОЗП = T_{розробки} \cdot N \cdot K_{кв},$$

де $T_{розробки}$ – витрати праці у чол./год;

N – погодинна ставка;

$K_{кв}$ – коефіцієнт кваліфікації програміста, приймається 0,75.

ОЗП складає:

$$ОЗП = 320 \cdot 55 \cdot 0,75 = 13200 \text{ грн.}$$

Відрахування на соціальні потреби встановлюються у відсотках від суми заробітної плати:

Отримані результати за (5.3) та (5.4) підсумовуються. Вони складають 16104

грн.та визначають основні прямі витрати.

5.3 Розрахунок накладних витрат

Накладні витрати враховують загальногосподарчі витрати по забезпеченню проведення роботи: витрати на опалення, електроенергію, амортизація, обладнання, зарплату

адміністративного персоналу та інше. Кожне підприємство визначає їх індивідуально, та зазвичай вони становлять 30-40% від суми прямих витрат на оплату праці:

На протязі усього терміну використання нової техніки підприємство щорічно витрачає певні кошти, пов'язані з її експлуатацією.

Експлуатаційні витрати на персональний комп'ютер визначаються протягом терміну розробки програмного засобу в залежності від вартості комп'ютеру. В експлуатаційні витрати входять:

- амортизаційні витрати на персональний комп'ютер і програмне забезпечення;
- загальногосподарські витрати (прибирання приміщення, охорона, оренда, комунальні послуги);
- витрати на електроенергію.

5.3.1 Розрахунок витрат на електроенергію

Витрати на електроенергію ($C_{\text{ел}}$) визначаються за формулою:

$$C_{\text{ел}} = P \cdot B \cdot T_{\text{розр}},$$

де P – потужність комп'ютера та допоміжних споживачів електричної енергії приймається 0,361 кВт/год;

B – вартість 1 кВт/год. Під час розробки диплому у 2019р. складає 1,68 грн[5];

$T_{\text{розр}}$ – вартість роботи з ЕВМ, прийнято рівним загальному робочому часу.

Отже у нашому випадку маємо:

$$C_{\text{ел}} = 0,361 \cdot 1,68 \cdot 320 = 194,08 \text{ грн.}$$

Витрати на матеріали ($C_{\text{м}}$) протягом всього терміну експлуатації приблизно 10% від вартості комп'ютеру. Вартість робочої станції приймається 29850 грн., термін експлуатації – 5 років. Отже, можна визначити ці витрати за період створення програмного засобу:

де $B_{\text{ком}}$ – вартість персонального комп'ютеру;

$N_{\text{д}}$ – кількість днів розробки програмного продукту;

$N_{\text{експ}}$ – термін експлуатації персонального комп'ютеру.

Витрати на матеріали визначаються так:

$$C_{\text{м}} = 29850 \cdot \frac{40}{5 \cdot 365} \cdot \frac{10}{100} = 65,43$$

Заробітна плата ремонтника ($C_{\text{рем}}$) визначена наступним чином: на ремонт 50 комп'ютерів потрібен один інженер-системотехнік. Його середньомісячна заробітна плата приймається 8000 грн. [6]. Тоді в перерахунку на один комп'ютер його заробітна плата складає:

де $C_{\text{рем}}$ – середньомісячна заробітна плата;

$N_{\text{ком}}$ – кількість комп'ютерів на одного ремонтника.

Заробітна плата ремонтника визначається так:

За статистикою витрати на комплектуючі вироби () для ремонту персонального комп'ютера складає 10% від його вартості за термін його експлуатації, тобто рівні витратам на витратні матеріали:

$$C_{\text{ком}} = C_{\text{вм}} = 65,43 \text{ грн.} \quad (5.9)$$

5.3.2 Розрахунок амортизаційних відрахувань

Амортизаційні відрахування на персональний комп'ютер (АПК) визначені з положення, що амортизаційний період в даний час дорівнює терміну морального старіння обчислювальної техніки і складає 5 роки. Отже, за 5 роки амортизаційні відрахування на персональний комп'ютер дорівнюють вартості комп'ютера. За період проектування амортизаційні відрахування складуть:

$$; \text{АПК} = 29850 \cdot \frac{40}{5 \cdot 365} = 654,25 \text{ грн.} \quad ($$

Амортизаційні відрахування на програмне забезпечення (АПЗ) залежать від його циклу заміни. Якщо прийняти термін морального старіння для Windows 5 років, Microsoft Excel 2010 за 3 роки та RAD Studio 10.2 за 10 років то амортизаційні відрахування на програмне забезпечення дорівнюють його вартості.

Для функціонування персонального комп'ютера використовувалася операційна система Windows Home 10, для написання програмного забезпечення програмне середовище RAD Studio 10.2, для завантаження та збереження даних використовується Microsoft Excel 2010.

$$\begin{aligned} \text{АПЗ}_{\text{W}} &= 3759 \cdot \frac{2}{5 \cdot 12} = 125,3 \text{ грн.} \\ \text{АПЗ}_{\text{RAD}} &= 141381,80 \cdot \frac{2}{10 \cdot 12} = 2356,36 \text{ грн.} \\ \text{АПЗ}_{\text{Excel}} &= 2547 \cdot \frac{2}{3 \cdot 12} = 141,5 \text{ грн.} \end{aligned}$$

Розрахунок амортизаційних відрахувань на програмне забезпечення зведений в табл. 5.2.

Таблиця 5.2 – Використовуване програмне забезпечення

Найменування програмного забезпечення	Вартість програмного забезпечення, грн	Джерело придбання	Амортизаційні відрахування, грн
---------------------------------------	--	-------------------	---------------------------------

Windows домашня	10	3759	https://soft.rozetka.com.ua/microsoft_kw9_00120/p4312887	125,3
RAD Studio 10.2		141381,80	http://store.embarcadero.ru/catalog/rubric/32	2356,36
Microsoft Excel 2010		2547	https://prom.ua/p647186480-kupit-excel-2010.html	141,5
Всього:		147687,8	-	2622,8

5.3.3 Комунальні послуги

Додаткові витрати : прибирання приміщень, охорона, аренда, комунальні послуги прийняти рівними 5790 гривень на місяць.

Оренду приміщень приймемо рівною 2692,3 гривень на місяць за 10 м² [7].

5.3.4 Зведений розрахунок накладних витрат

Результати розрахунку накладних витрат наведено в табл. 5.3, де враховані основні складові накладних витрат проекту:

$$C_{\text{експ}} = C_{\text{ел}} + C_{\text{вм}} + C_{\text{рем}} + \text{АПК} + \text{АПЗ} + C_{\text{ор}} + C_{\text{дод}} + C_{\text{Ком}};$$

$$C_{\text{експ}} = 194,08 + 65,43 + 340 + 654,25 + 2622,8 + 5384,6 + 5790 + 14881,16 + 65,43$$

$$= 15116,59 \text{ грн.}$$

Таблиця 5.3 – Розпис накладних витрат по проекту

№ п/п	Статті видатків	Сума, грн
1	Витрати на електроенергію	194,08
2	Вартість витратних матеріалів	65,43
3	Витрати на ремонт	340
4	Амортизація персонального комп'ютера	654,25
5	Амортизація програмного забезпечення	2622,8
6	Оренда приміщення	5384,6
7	Додаткові витрати	5790
8	Комплектуючі вироби	65,43
9	Всього накладних витрат	15116,59

5.3.5 Розрахунок витрат на створення програмного продукту

Таким чином, витрати на створення програмного комплексу в рамках виконання проекту складають:

$$- C_{\text{розробки}} = \text{ОЗП} + C_{\text{соц}} + C_{\text{накл}} + C_{\text{експ}}; \quad ($$

$$- C_{\text{розробки}} = 13200 + 2904 + 5153,28 + 15116,59 = 36373,87 \text{ грн.}$$

Розрахунок витрат зведено у табл. 5.4.

Таблиця 5.4 – Кошторис витрат на розробку програмного засобу

№ п/п	Найменування витрат	Сума, грн
1	Основна заробітна плата	13200
2	Відрахування на соціальні потреби	2904
3	Накладні витрати	5153,28
4	Експлуатаційні витрати	15116,59
5	Всього	36373,87

За отриманими значеннями техніко-економічних показників проекту складено кошторис витрат на розробку сучасного програмного забезпечення для моделювання нечітких часових рядів. За результатами розрахунків, приблизна вартість розробки складає 36373,87 грн.

6 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ

Усі стадії та етапи розробки приведені у табл.6.1.

Таблиця 6.1– Етапи розробки та строки

Стаді розробки	Етап розробки	Термін
Технічне завдання (ТЗ)	Постановка задачі, збір початкових матеріалів	15.12.2021 – 22.12.2021
	Розробка структур вхідних та вихідних даних	15.12.2021 – 22.12.2021
	Визначення вимог до програми	15.12.2021 – 22.12.2021
	Узгодження та затвердження технічного завдання	15.12.2021 – 22.12.2021
Робочий проект	Програмування та відладка програми	15.12.2021 – 22.12.2021
	Тестування програми	15.12.2021 – 22.12.2021
Впровадження	Розробка, узгодження та затвердження програмної документації	15.12.2021 – 22.12.2021

7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙОМУ

Контроль здійснюється за допомогою виконання набору тестів з метою знаходження помилок в програмі та його специфікації. Контроль виконання роботи забезпечується головним керівником розробки.

Прийом програми здійснюється уповноваженою комісією.

СПИСОК ЛІТЕРАТУРИ

1. ДСанПіН 3.3.2-007-98. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин [Текст] / Постанова Головного державного санітарного лікаря України від 10 грудня 1998 р. № 7 – К., 1998.
2. Основи стандартизації програмних систем [Текст]: методичні вказівки до дипломного проектування та лабораторних робіт / уклад.: Ю. М. Івченко, В. І. Шинкаренко, В. Г. Івченко; Дніпропетр. нац. ун-т залізн. трансп. ім. акад. В. Лазаряна. – Д.: Вид-во Дніпропетр. нац. ун-ту залізн. трансп. ім. акад. В. Лазаряна, 2009. – 38 с.
3. Медведкова І. В. Порівняльний аналіз методів оцінки вартості проектів з розробки програмного забезпечення [Електронний ресурс] /І. В. Медведкова, А. А. Іванов [Москва: національний дослідницький університет МІФІ] – №7. – Режим доступу cyberleninka.ru (дата звернення 15.04.2019 р.).
4. Інженер-програміст: середня зарплата в Україні [Електронний ресурс] – Режим доступу:<http://www.wikipage.com.ua/1x4306.html>tech.informator.ua
5. Тарифи на електроенергію в 2019 році [Електронний ресурс] – Режим доступу: www.prostobank.ua
6. Комп'ютерник: середня зарплата в Україні [Електронний ресурс] – Режим доступу: <https://www.work.ua>
7. Статистика цін на оренду квартир в Дніпропетровську [Електронний ресурс] – Режим доступу: 100realty.ua
8. Бутстреп, малі вибірки, застосування в аналізі даних [Електронний ресурс] – Режим доступу: <http://statistica.ru/theory/metod-butstrepa-i-ego-primenenie-v-sovremennom-analize-dannykh/>

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАТВЕРДЖЕНО
1116130. 01165-01 ІЗ 01 ЛЗ

Дослідження моделей антиперсистентних часових послідовностей
Керівництво користувача. Керівництво з моделювання нечіткої часової
послідовності

1116130. 01165-01 ІЗ 01

Аркушів 20

АНОТАЦІЯ

Документ 1116130.01165-01 ІЗ «Дослідження моделей антиперсистентних часових послідовностей» Керівництво користувача. Керівництво з моделювання часової послідовності, що входить до складу програмної документації до дипломного проекту.

У даному документі представлене призначення та умови застосування програми, підготовка до роботи, опис операцій, аварійні ситуації, рекомендації щодо засвоєння.

ЗМІСТ

Вступ.....	27
1. Призначення та умови застосування.....	28
2. Підготовка до роботи.....	29
3. Опис операцій.....	30
4. Аварійні ситуації.....	34
5. Рекомендації щодо засвоєння	35
6. Повідомлення користувачу	37

ВСТУП

Програмний комплекс надає можливість користувачеві обробляти процеси, будь якого виду, які представлені у вигляді часового ряду. Для цього програмний комплекс надає такі можливості прогнозувати майбутні значення часового ряду, побудова антиперсистентної моделі на основі вхідного часового ряду, розмноження часового ряду методом бутстрепа, побудова часових рядів на графіку з можливістю вибору перегляду значень величин, значення величин t , модель інтерполяції, завантаження та збереження моделей часових рядів з файлу Microsoft Office Excel.

Для користування програмним комплексом користувач повинен володіти базовими знаннями користування персональним комп'ютером. Для того щоб користувач міг використовувати програмний комплекс у повному обсязі, необхідно ознайомитися з програмною документацією опис програми та керівництво користувача.

1. ПРИЗНАЧЕННЯ ТА УМОВИ ЗАСТОСУВАННЯ

Програмний комплекс призначена для автоматичної обробки часових рядів різноманітних технологічних, виробничих, інформаційних, лікувальних та інших процесів, швидкому порівнянні отриманих моделей, а також прогнозування за допомогою створеного програмного комплексу.

Умови для використання програмного комплексу такі:

- операційна система Winwos;
- інстальований Microsoft Office Excel;
- забезпечити контроль вхідних та вихідних даних, описані в описі програми;
- процесор AMD Ryzen 2200g;
- оперативна пам'ять – від 1 ГБ;
- клавіатура;
- комп'ютерна «миша»;
- CD/DVD-RW привід;
- USB-роз'єм.

2. ПІДГОТОВКА ДО РОБОТИ


На носії знаходяться виконавчий файли програмного комплексу та програмна документація, які можуть бути перенесені на комп'ютер і використовуватися за призначенням.

Для виклику програмного комплексу необхідно виконати `an-pers.exe` файл, після чого програмний комплекс може бути використано для обробки антиперсистентних часових рядів.

Для перевірки програмного комплексу необхідно запустити його і провести перевірку всіх функцій. Для початкової перевірки достатньо завантажити початковий часовий ряд та побудувати модель антиперсистентного ряду і зберегти інформацію.

3. ОПИС ОПЕРАЦІЙ

- Після запуску програмного комплексу, з'явиться головне вікно зображене на рис. 3.1.



27.12.2014	11,9
15.03.2015	12,5
20.06.2015	10,8
15.07.2015	9,8
22.07.2015	10,1
29.07.2015	9,7
07.09.2015	12,6
21.09.2015	13
15.10.2015	15,8
31.01.2016	7,5
12.04.2016	11
18.06.2016	6
06.01.2017	12,3
06.03.2017	9,7

- Рисунок 3.1 – Головне вікно програмного комплексу

З головного вікна програмного комплексу виконуються функції, описані нижче:

Функція завантаження. Натисніть на вкладку «Файл» та оберіть «Завантажити», після чого відкриється вікно Microsoft Office Excel зображено на рис. 3.2, далі засобами Microsoft Office Excel відкрити файл де знаходиться нечіткий часовий ряд, натиснути кнопку завантажити в програмному комплексі зображено на рис. 3.3, дані відобразяться в головному вікні програмного комплексу.

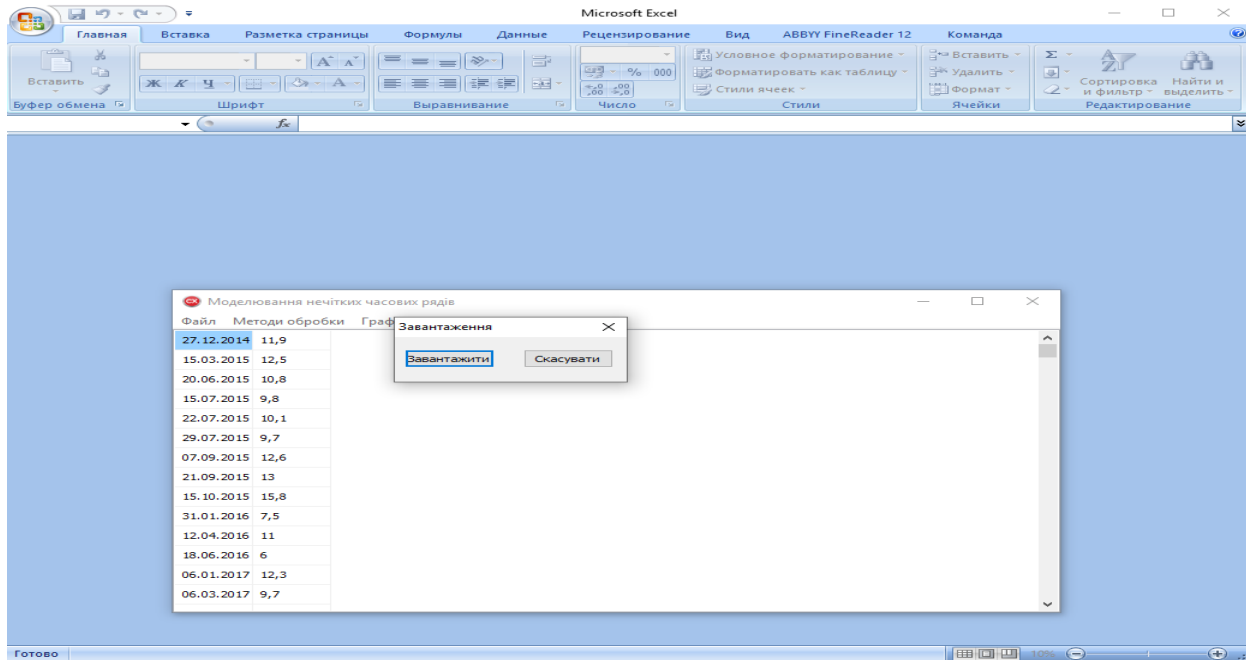


Рисунок 3.2 – Виконання завантаження

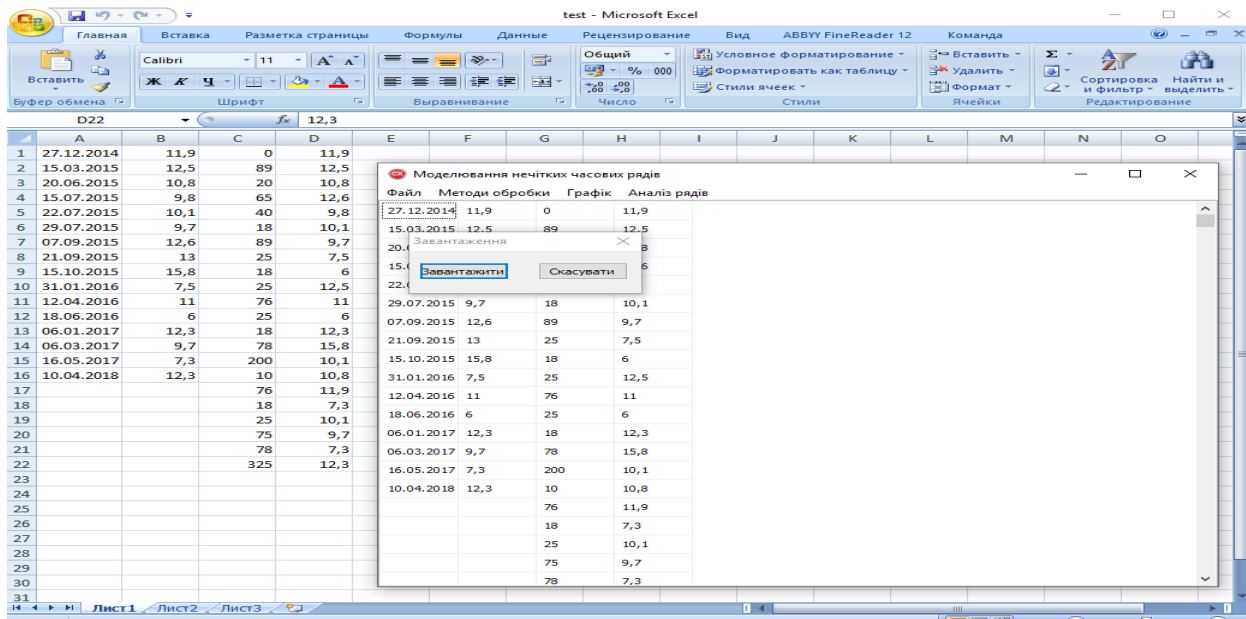


Рисунок 3.3 – Завантаження даних з Excel

Функція збереження. Натисніть на вкладку «Файл» та оберіть «Зберегти», після чого відкриється вікно Microsoft Office Excel зображено на рис. 3.4, в якому будуть розміщені дані з програмного комплексу, засобами Microsoft Office Excel зберегти дані.

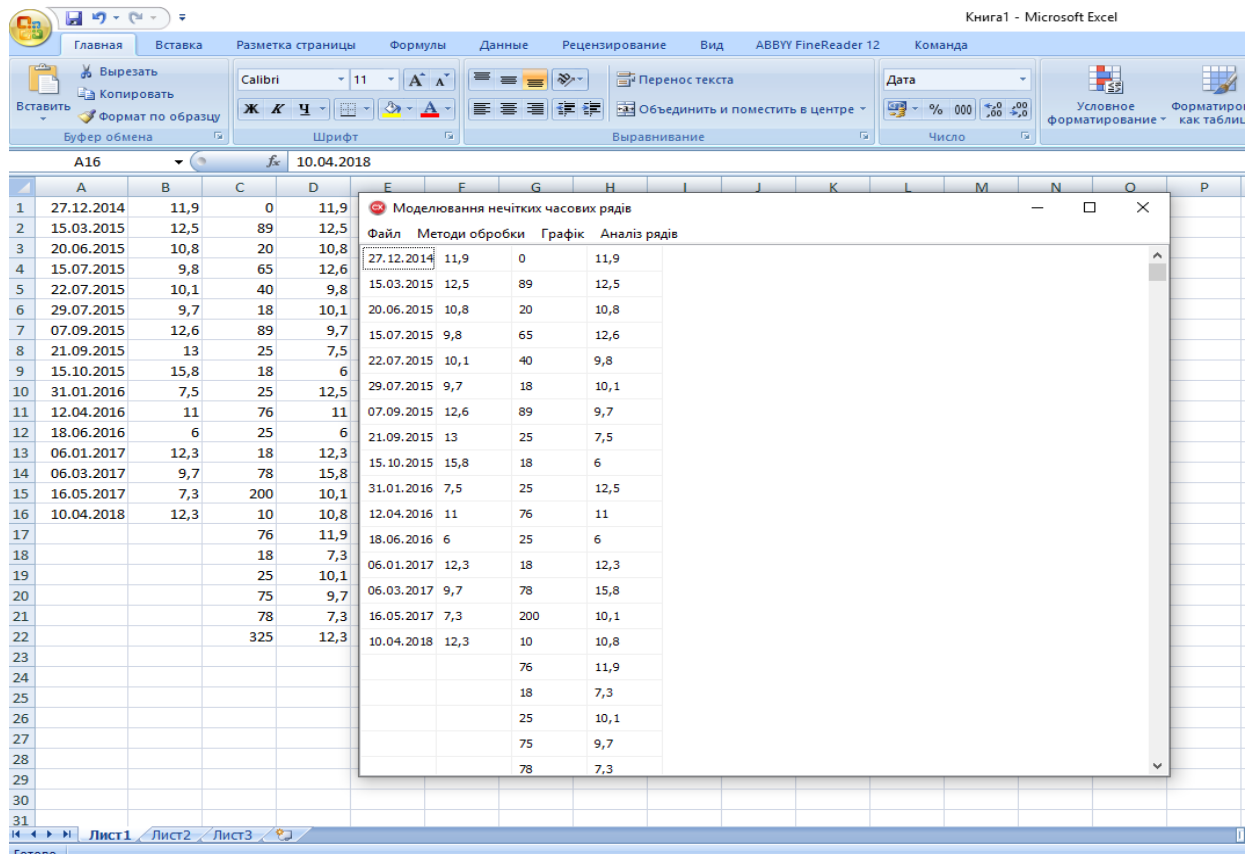


Рисунок 3.4 – Функція збереження

Функція розмноження часового ряду бутстрепом. Натиснути на вкладку «Методи обробки» та обрати «розмноження НЧР бутстрепом», після чого відкриється вікно бутстрепа, ввести параметр фіксовані точки та розмір нового нечіткого часового ряду, далі виділяємо мишкою вхідні ряди та натискаємо на кнопку «Розмноження ряду», налаштування бутстрепа зображено на рис. 3.10. Також слід відмітити на отриманий розмножений ряд відразу виконається функція багатокрокового методу, за замовчування параметри багатокрокового методу налаштовані. Отриманий результат зображено на рис. 3.11.

Функція побудови нечіткого часового ряду на основі декількох рівнів. Натиснути на вкладку «Аналіз рядів», після чого відкриється вікно аналіз рядів, ввести параметр кількість рівнів, далі виділяємо мишкою вхідні ряди та натискаємо на кнопку «Побудувати ряд», налаштування аналізу ряду зображено на рис. 3.12. Також слід відмітити на отриманий розмножений ряд

відразу виконається функція багатокрокового методу, за замовчування параметри багатокрокового методу налаштовані. Отриманий результат зображено на рис. 3.13.

Функція додання ряду. Натиснути на пункт «Графік» та обрати «додати ряд» після чого з'явиться вікно додання ряду 3.15, ввести ім'я нового ряду, далі виділяємо мишкою вхідні ряди та натискаємо на кнопку «Додати графік», результат зображено на рис. 3.16.

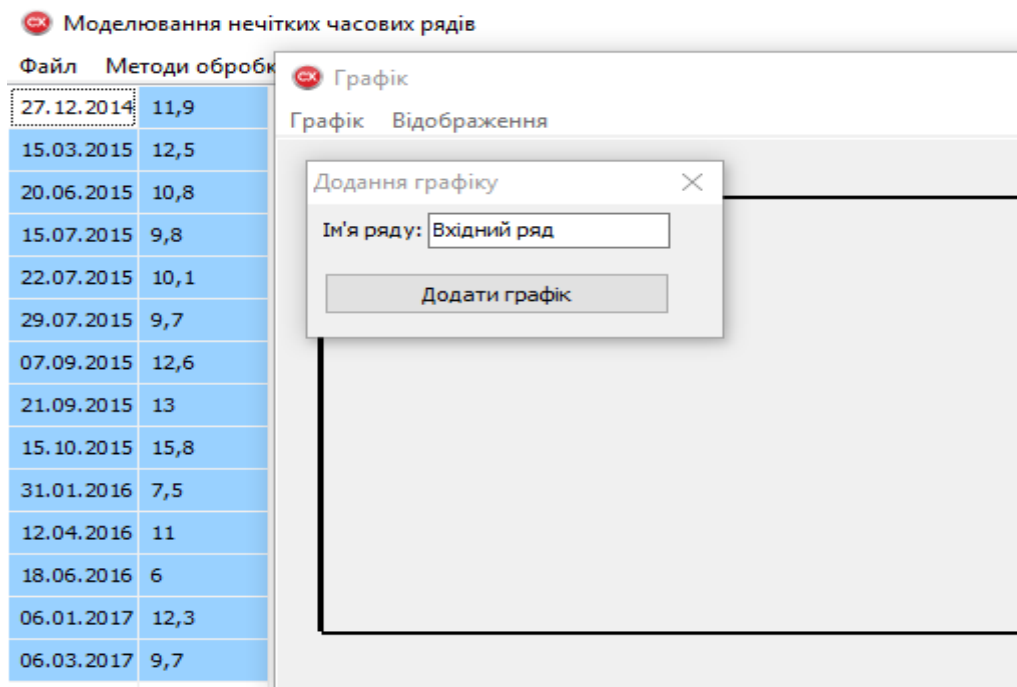


Рисунок 3.15 – Налаштування додання графіку

Рисунок 3.16 – Доданий ряд на графік

Функція очистити графік. Натиснути на пункт «Графік» та обрати «Очистити графік» після чого всі графіки будуть видалені, приклад роботи зображено на рис. 3.17.

4. АВАРІЙНІ СИТУАЦІЇ

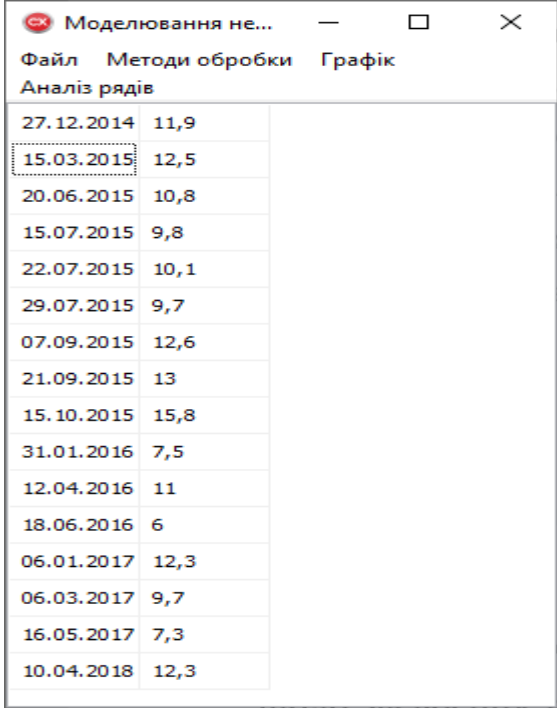
При недотриманні умов технологічного процесу будуть виводитися відповідні повідомлення або значення будуть встановлені автоматично.

При відмові носія програмні данні будуть втрачені, якщо їх не зберегти на додатковому носії.

Програмний комплекс не має захисту від зломів хакерів оскільки працює з даними з файлу після завершення роботи дані вивантажуються та зберігаються у файл Microsoft Office Excel. При необхідності захистити дані Microsoft Office Excel має власну систему захисту даних, тому захист програмному комплексу не потрібен.

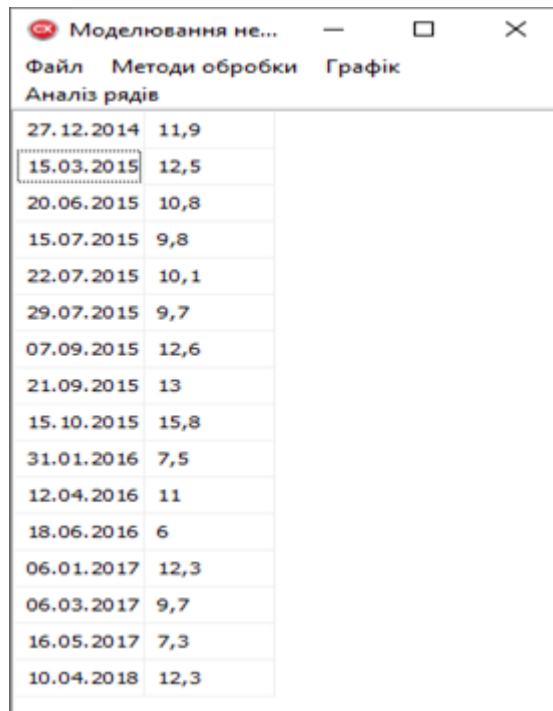
5. РЕКОМЕНДАЦІЇ ЩОДО ЗАСВОЄННЯ

- Перевіряйте значення котрі вводяться до полів та дотримуйтесь інструкції користування описаних в описі операцій. За замовчуванням завантажується тестовий вхідний ряд зображено на рис. 5.1, котрий одразу можливо обробити



Аналіз рядів	
27.12.2014	11,9
15.03.2015	12,5
20.06.2015	10,8
15.07.2015	9,8
22.07.2015	10,1
29.07.2015	9,7
07.09.2015	12,6
21.09.2015	13
15.10.2015	15,8
31.01.2016	7,5
12.04.2016	11
18.06.2016	6
06.01.2017	12,3
06.03.2017	9,7
16.05.2017	7,3
10.04.2018	12,3

Рисунок 5.1 – Головне вікно



The image shows a screenshot of a software application window. The title bar contains the text 'Моделювання не...' followed by standard window control icons (minimize, maximize, close). Below the title bar is a menu bar with the following items: 'Файл', 'Методи обробки', 'Графік', and 'Аналіз рядів'. The main area of the window contains a table with two columns. The first column lists dates, and the second column lists numerical values. The date '15.03.2015' is highlighted with a dashed border.

Дата	Значення
27.12.2014	11,9
15.03.2015	12,5
20.06.2015	10,8
15.07.2015	9,8
22.07.2015	10,1
29.07.2015	9,7
07.09.2015	12,6
21.09.2015	13
15.10.2015	15,8
31.01.2016	7,5
12.04.2016	11
18.06.2016	6
06.01.2017	12,3
06.03.2017	9,7
16.05.2017	7,3
10.04.2018	12,3

Рисунок 5.2 – Побудова моделі

6.ПОВІДОМЛЕННЯ КОРИСТУВАЧУ

Для ефективної та правильної роботи необхідно читати повідомлення, котрі допоможуть при помилкових введеннях значеннях (рис. 5.1-5.3)

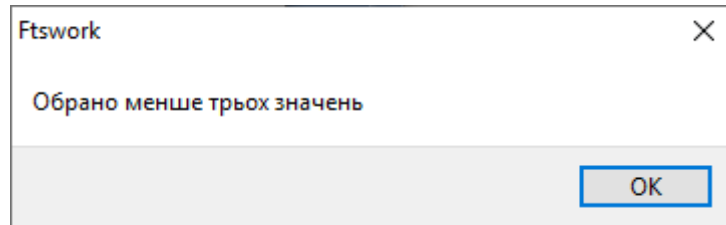


Рисунок 5.1 – При обрані недостатнього розміру ряду

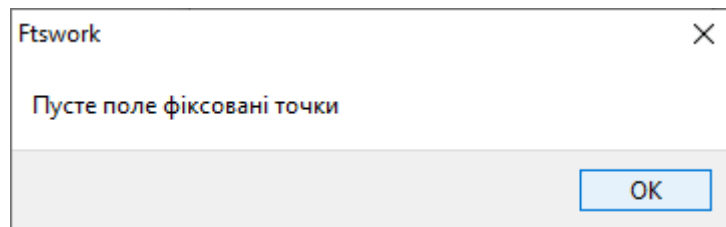


Рисунок 5.2 – При пропуску полі

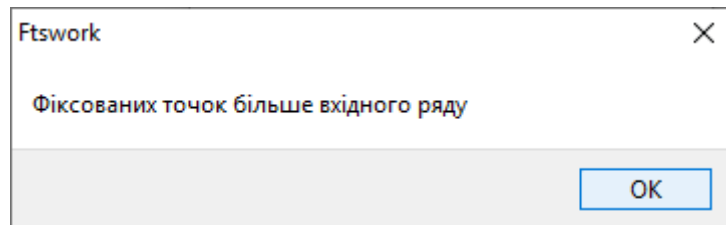


Рисунок 5.3 – При введенні не коректних пропорцій

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАТВЕРДЖЕНО
1116130.01165-01 12 01-ЛЗ

Дослідження моделей антиперсистентних часових послідовностей

Текст програми

1116130.01165-01 12 01

Аркушів 20

2021

АНОТАЦІЯ

Документ 1116130.01165-01 12 «Дослідження моделей антиперсистентних часових послідовностей»» Текст програми, що входить до складу програмної документації до дипломного проекту.

У даному документі представлена структура програми, текст програми.

3
1116130. 01165-01 12 01
ЗМІСТ

1 Структура програми.....	4
2 Текст програми	5

1116130. 01165-01 12 01
СТРУКТУРА ПРОГРАМИ

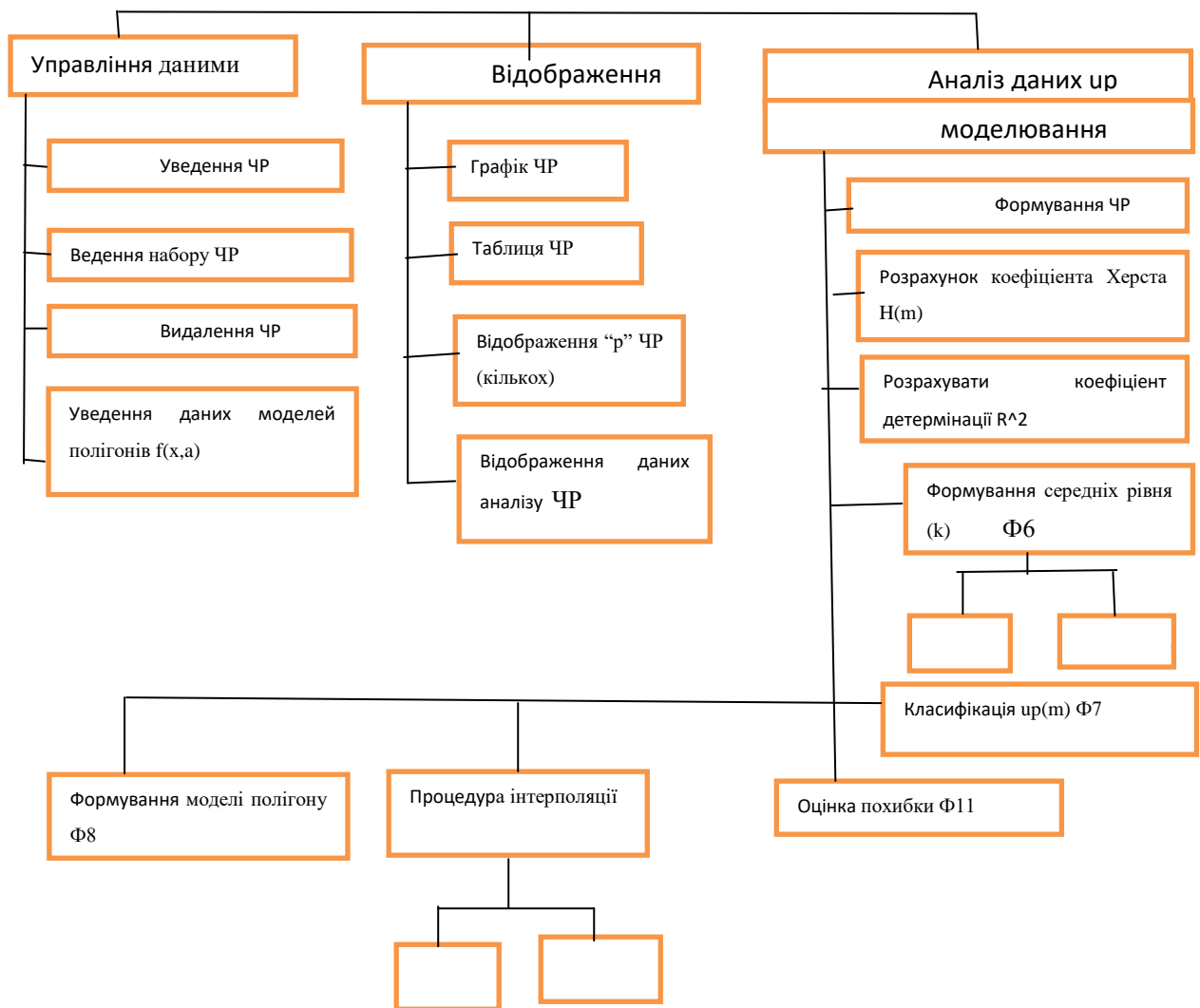


Рисунок Д.1 – Функціональні можливості програмного комплексу

Програмний комплекс класифікація:

- 1) управління даними, введення ЧР, ведення ЧР, видалення ЧР, введення даних моделей полігонів,
- 2) відображення даних: графік ЧР, таблиця ЧР, відображення ЧР, відображення даних аналізу ЧР,
- 3) аналіз даних ЧР та моделювання, формування ЧР моделі, розрахунок коефіцієнта Херста, розрахунок коефіцієнта детермінації, формування середніх рівня, класифікація ЧР, формування моделі полігону, процедури інтерполяції, оцінка похибки

ТЕКСТ ПРОГРАМИ

main.h

```
//-----

#ifndef mainH
#define mainH
//-----
#include <System.Classes.hpp>
#include <Vcl.Controls.hpp>
#include <Vcl.StdCtrls.hpp>
#include <Vcl.Forms.hpp>
#include <Vcl.Dialogs.hpp>
#include <Vcl.Grids.hpp>
#include <Vcl.Menus.hpp>
#include "Excel_XP_srvr.h"
#include <Vcl.OleServer.hpp>
//-----
class TForm1 : public TForm
{
__published:      // IDE-managed Components
    TStringGrid *SG;
    TMainMenu *MainMenu1;
    TMenuItem *N1;
    TMenuItem *N2;
    TMenuItem *N3;
    TMenuItem *N4;
    TMenuItem *N5;
    TMenuItem *N6;
    TMenuItem *N7;
    TExcelApplication *EA;
    TExcelWorksheet *EW;
    TMenuItem *N8;

    void __fastcall N2Click(TObject *Sender);
    void __fastcall N5Click(TObject *Sender);
    void __fastcall N6Click(TObject *Sender);
    void __fastcall SGColumnMoved(TObject *Sender, int FromIndex, int ToIndex);
    void __fastcall N7Click(TObject *Sender);
    void __fastcall FormCreate(TObject *Sender);
    void __fastcall N3Click(TObject *Sender);
    void __fastcall N8Click(TObject *Sender);

private:      // User declarations
public:      // User declarations
    int left;
    int right;
    int top;
    int bottom;

    Variant app;
    __fastcall TForm1(TComponent* Owner);
};
//-----
extern PACKAGE TForm1 *Form1;
//-----
#endif
```

main.cpp

```
//-----

#include <vcl.h>
#pragma hdrstop

#include "main.h"
#include "MultiStep.h"
#include "ftsms.h"
#include "ftschart.h"
#include <DateUtils.hpp>
#include "analizseries.h"
#include <ComObj.hpp>
```

```

#include <utilcls.h>
#include "exelload.h"
#include "bootstrep.h"
//-----
#pragma package(smart_init)
#pragma link "Excel_XP_srvr"
#pragma resource "*.dfm"
TForm1 *Form1;

//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
//-----
//Обробка кнопки завантаження
void __fastcall TForm1::N2Click(TObject *Sender)
{
    app = CreateOleObject("Excel.Application");
    app.OlePropertySet("Visible",true);
    Form6->Caption="Завантаження";
    Form6->Button1->Caption="Завантажити";
    Form6->Visible=true;
    ;
}
//-----
//Обробка кнопки багато кроковий метод
void __fastcall TForm1::N5Click(TObject *Sender)
{
    left=SG->Selection.Left;
    right=SG->Selection.Right;
    top=SG->Selection.Top;
    bottom=SG->Selection.Bottom;
    Form2->Visible=true;
}
//-----
//Обробка кнопки графік
void __fastcall TForm1::N6Click(TObject *Sender)
{
    Form3->Visible=true;
}
//-----
//Обробка розмноження нчр бутстрепом
void __fastcall TForm1::N7Click(TObject *Sender)
{
    Form5->Visible=true;
}
//-----
//створення форми та завантаження стартового ряду
void __fastcall TForm1::FormCreate(TObject *Sender)
{
    TStringList *list_date=new TStringList();
    list_date->LoadFromFile("1945.txt");
    SG->ColCount=2;
    SG->RowCount=list_date->Count/2;
    for(int y=0;y<SG->ColCount;y++)
    for(int x=0;x<SG->RowCount;x++)
    SG->Cells[y][x]=list_date->Strings[x+y*SG->RowCount];
}
//-----
//Обробка кнопки збереження
void __fastcall TForm1::N3Click(TObject *Sender)
{
    Variant app = CreateOleObject("Excel.Application");
    app.OlePropertySet("Visible",true);
    app.OlePropertyGet("WorkBooks").OleProcedure("Add");
    for(int x=0;x<Form1->SG->RowCount;x++)
    for(int y=0;y<Form1->SG->ColCount;y++)
    {
        if(SG->Cells[y][x]!="")
        {
            Form1->SG->Cells[y][x]=Form1->SG->Cells[y][x]+" ";
        }
        app.OlePropertyGet("WorkBooks",1).OlePropertyGet("Worksheets",1).OlePropertyGet("Cells",x+1,y+1).OlePropertySet("Value",
        Form1->SG->Cells[y][x].c_str());
    }
}

```

```

    }
}
//-----
//Обробка аналіз ряду
void __fastcall TForm1::N8Click(TObject *Sender)
{
    left=SG->Selection.Left;
    right=SG->Selection.Right;
    top=SG->Selection.Top;
    bottom=SG->Selection.Bottom;
    Form7->Visible=true;
}
//-----

```

ftsms.h

```

//-----

#ifndef ftsmsH
#define ftsmsH
//-----
#include <System.Classes.hpp>
#include <Vcl.Controls.hpp>
#include <Vcl.StdCtrls.hpp>
#include <Vcl.Forms.hpp>
//-----
class TForm2 : public TForm
{
__published:        // IDE-managed Components
    TLabel *Label3;
    TEdit *forepoint;
    TButton *Button1;
    TEdit *kappa;
    TLabel *Label5;
    void __fastcall Button1Click(TObject *Sender);
private:            // User declarations
public:              // User declarations
    void buildfts();
    void analizseries(int coun);
    void bootstrep(int fiks, int count);
    __fastcall TForm2(TComponent* Owner);
};
//-----
extern PACKAGE TForm2 *Form2;
//-----
#endif

```

ftsms.cpp

```

//-----

#include <vcl.h>
#pragma hdrstop
#include "MultiStep.h"
#include "ftsms.h"
#include "main.h"
#include "DateConvert.h"
#include "editftschart.h"
#include "ftschart.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm2 *Form2;
MultiStep *calculateforecast(float *mas, int size, int forepoint, float kappa);
//-----
__fastcall TForm2::TForm2(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void TForm2::analizseries(int count)
{
    Form1->left=Form1->SG->Selection.Left;
    Form1->right=Form1->SG->Selection.Right;
    Form1->top=Form1->SG->Selection.Top;
    Form1->bottom=Form1->SG->Selection.Bottom;
    float *mas=new float[Form1->bottom-Form1->top+1];

    int coun=count-1;
}

```

```

float sum;

DateConvert *dconvert,*ddecode;;
if(Form1->right==Form1->left)
{

    Form1->SG->ColCount=Form1->SG->ColCount+1;
    Form1->SG->Refresh();

    bool flag=false;

    if(Form1->SG->Cells[Form1->right][Form1->top].Pos(".")!=0)
    {
        flag=true;
        float *massum=new float[Form1->bottom-Form1->top];
        dconvert=new DateConvert(Form1->SG, Form1->bottom+1,Form1->top,Form1->left);
        for(int x=0;x<=Form1->bottom-Form1->top+1;x++)
            mas[x]=dconvert->Getstep(x);

        for(int x=Form1->top+coun;x<=Form1->bottom;x++)
        {
            sum=0;
            for(int y=x-coun-Form1->top; y<=x;y++)
                sum+=mas[y];
            sum=sum/(coun+1);
            massum[x-coun]=sum;
        }
        for(int x=Form1->top+coun;x<=Form1->bottom;x++)
            Form1->SG->Cells[Form1->SG->ColCount-1][x]=FloatToStr(massum[x-coun]);
    }
    else
    for(int x=Form1->top+coun;x<=Form1->bottom;x++)
    {
        sum=0;
        for(int y=x-coun; y<=x;y++)
            sum+=StrToFloat(Form1->SG->Cells[Form1->right][y]);
        sum=sum/(coun+1);
        Form1->SG->Cells[Form1->SG->ColCount-1][x]=FloatToStr(sum);
    }

    if(flag)
    {
        ddecode=new          DateConvert(Form1->SG,Form1->bottom+1,Form1->top+coun,Form1->SG-
>ColCount-1,Form1->SG->Cells[Form1->right][Form1->top+coun-1]);
        Form1->SG->ColCount+=1;
        for(int x=Form1->top+coun;x<=Form1->bottom;x++)
            Form1->SG->Cells[Form1->SG->ColCount-1][x]=ddecode->Getdate(x-Form1->top-
coun);

        delete ddecode;
        delete dconvert;
    }
    Form1->left=Form1->SG->ColCount-1;
    Form1->right=Form1->SG->ColCount-1;
    Form1->top=Form1->top+coun;

}
else
{

    if(Form1->SG->Cells[Form1->left][Form1->top].Pos(".")!=0)
    {
        Form1->SG->ColCount+=1;

        dconvert=new DateConvert(Form1->SG, Form1->bottom+1,Form1->top,Form1->left);
        for(int x=0;x<=Form1->bottom-Form1->top+1;x++)
            mas[x]=dconvert->Getstep(x);
        delete dconvert;
        for(int x=Form1->top+coun;x<=Form1->bottom;x++)
        {
            sum=0;
            for(int y=x-coun-Form1->top; y<=x;y++)
                sum+=mas[y];
            sum=sum/(coun+1);

```


1116130.01165-01 12 01

```

Form1->SG->Cells[Form1->SG->ColCount-1][x]=FloatToStr(sum);
}
ddecode=new DateConvert(Form1->SG,Form1->bottom+1,Form1->top+coun,Form1->SG-
>ColCount-1,Form1->SG->Cells[Form1->left][Form1->top+coun-1]);
Form1->SG->ColCount+=1;
for(int x=Form1->top+coun;x<=Form1->bottom;x++)
    Form1->SG->Cells[Form1->SG->ColCount-1][x]=ddecode->Getdate(x-Form1->top-
coun);

    delete ddecode;
}
Form1->left=Form1->SG->ColCount-1;
Form1->SG->ColCount+=1;
float *massum=new float[Form1->bottom-Form1->top];
for(int x=Form1->top+coun;x<=Form1->bottom;x++)
{
    sum=0;
    for(int y=x-coun;y<=x;y++)
        sum+=StrToFloat(Form1->SG->Cells[Form1->right][y]);
    sum=sum/(coun+1);
    massum[x-coun]=sum;
}
for(int x=Form1->top+coun;x<=Form1->bottom;x++)
    Form1->SG->Cells[Form1->SG->ColCount-1][x]=FloatToStr(massum[x-coun]);

Form1->right=Form1->SG->ColCount-1;
Form1->top=Form1->top+coun;
}
delete mas;
}
//-----
void TForm2::buildfts()
{

    Form4->namechart->Text="BP";
    Form3->clearchart();

    Form4->addchart();
    if(forepoint->Text=="")
        forepoint->Text="3";
    if(kappa->Text=="")
        kappa->Text="0";
    float *mas=new float[Form1->bottom-Form1->top+StrToInt(forepoint->Text)];
    DateConvert *dconvert,*ddecode;
    if(Form1->SG->RowCount<=Form1->bottom+StrToInt(forepoint->Text)+1)
        Form1->SG->RowCount=Form1->SG->RowCount+(Form1->bottom+StrToInt(forepoint->Text)+1-Form1-
>SG->RowCount);

    MultiStep* forecast=new MultiStep[Form1->bottom-Form1->top];
    MultiStep* fors;
    if(Form1->right==Form1->left)
    {
        Form3->view->Enabled=false;

        bool flag=false;
        Form1->SG->ColCount+=1;
        if(Form1->SG->Cells[Form1->right][Form1->top].Pos(".")!=0)
        {
            flag=true;
            dconvert=new DateConvert(Form1->SG, Form1->bottom+1,Form1->top,Form1->left);
            for(int x=0;x<=Form1->bottom-Form1->top+1;x++)
                mas[x]=dconvert->Getstep(x);
        }
        else
        {
            for(int x=Form1->top;x<=Form1->bottom;x++)
                mas[x-Form1->top]=StrToFloat(Form1->SG->Cells[Form1->right][x]);
        }
        fors=calculateforecast(mas,Form1->bottom-Form1->top+StrToInt(forepoint->Text),StrToInt(forepoint-
>Text),StrToFloat(kappa->Text));
        for(int x=Form1->top+3;x<=Form1->bottom+StrToInt(forepoint->Text)+1;x++)
        {
            Form1->SG->Cells[Form1->SG->ColCount-1][x]=FloatToStr(fors[x-Form1->top-3].Getforecast());
        }
        if(flag)
        {

```

```

Form1->SG->ColCount+=1;
ddecode=new DateConvert(Form1->SG,Form1->bottom+StrToInt(forepoint->Text)+1,Form1-
>top+3,Form1->SG->ColCount-2,Form1->SG->Cells[Form1->right][Form1->top+2]);

for(int x=Form1->top+3;x<=Form1->bottom+StrToInt(forepoint->Text)+1;x++)
    Form1->SG->Cells[Form1->SG->ColCount-1][x]=ddecode->Getdate(x-Form1->top-3);
delete ddecode;
delete dconvert;
}
Form1->left=Form1->SG->ColCount-1;
Form1->right=Form1->SG->ColCount-1;
}
else
{
    Form3->view->Enabled=true;
    Form3->posseries[0]=Form1->top;
    Form3->posseries[1]=Form1->bottom;
    Form3->posseries[2]=Form1->left;
    Form3->posseries[3]=Form1->right;
    Form3->posseries[4]=Form1->top+3;
    Form3->posseries[5]=Form1->bottom+StrToInt(forepoint->Text);

    Form3->separabel->Checked=true;
    Form1->SG->ColCount+=3;
    dconvert=new DateConvert(Form1->SG, Form1->bottom+1,Form1->top,Form1->left);
    for(int x=0;x<=Form1->bottom-Form1->top+1;x++)
        mas[x]=dconvert->Getstep(x);
    fors=calculateforecast(mas,Form1->bottom-Form1->top+StrToInt(forepoint->Text),StrToInt(forepoint-
>Text),StrToFloat(kappa->Text));
    for(int x=Form1->top+3;x<Form1->bottom+StrToInt(forepoint->Text)+1;x++)
    {
        Form1->SG->Cells[Form1->SG->ColCount-3][x]=FloatToStr(fors[x-Form1->top-3].Getforecast());
    }
    ddecode=new DateConvert(Form1->SG,Form1->bottom+StrToInt(forepoint->Text)+1,Form1->top+3,Form1-
>SG->ColCount-3,Form1->SG->Cells[Form1->left][Form1->top+2]);
    for(int x=Form1->top+3;x<Form1->bottom+StrToInt(forepoint->Text)+1;x++)
        Form1->SG->Cells[Form1->SG->ColCount-2][x]=ddecode->Getdate(x-Form1->top-3);
    for(int x=Form1->top;x<=Form1->bottom;x++)
        mas[x-Form1->top]=StrToFloat(Form1->SG->Cells[Form1->right][x]);
    fors=calculateforecast(mas,Form1->bottom-Form1->top+StrToInt(forepoint->Text),StrToInt(forepoint-
>Text),StrToFloat(kappa->Text));
    for(int x=Form1->top+3;x<Form1->bottom+StrToInt(forepoint->Text)+1;x++)
    {
        Form1->SG->Cells[Form1->SG->ColCount-1][x]=FloatToStr(fors[x-Form1->top-3].Getforecast());
    }
    Form1->left=Form1->SG->ColCount-2;
    Form1->right=Form1->SG->ColCount-1;
    Form3->posseries[6]=Form1->left;
    Form3->posseries[7]=Form1->right;
}
delete forecast;
delete mas;
if(fors!=NULL)
    delete fors;

Form1->top=Form1->top+3;
Form1->bottom=Form1->bottom+StrToInt(forepoint->Text);
Form4->namechart->Text="BP модель";
Form4->addchart();
Form4->namechart->Text="";
Form3->Visible=true;

}
//-----
void __fastcall TForm2::Button1Click(TObject *Sender)
{
    Form1->left=Form1->SG->Selection.Left;
    Form1->right=Form1->SG->Selection.Right;
    Form1->top=Form1->SG->Selection.Top;
    Form1->bottom=Form1->SG->Selection.Bottom;
    bool flag=true;
    int x,y;
    y=Form1->left;
    x=Form1->right;
    x=x-y;
    if(x>1)
    {
        ShowMessage("Обрано більше двох рядів");
    }
}

```

```

        flag=false;
    }
    y=Form1->top;
    x=Form1->bottom;
    x=x-y;
    if(x<1)
    {
        ShowMessage("Обрано менше трьох значень");
        flag=false;
    }
    if(flag)
        buildfts();
}
//-----
void TForm2::bootstrep(int fiks, int count)
{
    float *mas=new float[Form1->bottom-Form1->top+1];
    float *masbootstrep=new float[count+1];
    DateConvert *dconvert,*ddecode;
    if(Form1->SG->RowCount<=count+1)
        Form1->SG->RowCount=count+1;
    if(Form1->right==Form1->left)
    {
        bool flag=false;
        Form1->SG->ColCount+=1;
        if(Form1->SG->Cells[Form1->right][Form1->top].Pos(".")!=0)
        {
            flag=true;
            dconvert=new DateConvert(Form1->SG, Form1->bottom+1,Form1->top,Form1->left);
            for(int x=0;x<=Form1->bottom-Form1->top+1;x++)
                mas[x]=dconvert->Getstep(x);
            delete dconvert;
        }
        else
        {
            for(int x=Form1->top;x<=Form1->bottom;x++)
                mas[x-Form1->top]=StrToFloat(Form1->SG->Cells[Form1->right][x]);
        }
        for(int x=0;x<fiks;x++)
            masbootstrep[x]=mas[x];
        for(int x=count-fiks;x<count;x++)
            masbootstrep[x]=mas[x-(count-fiks)+Form1->bottom+1-fiks];
        for(int x=Form1->top+fiks;x<count-fiks;x++)
        {
            int index=Form1->top+rand()/32767.0*(Form1->bottom-Form1->top);
            masbootstrep[x]=mas[index];
        }
        for(int x=Form1->top;x<count;x++)
        {
            Form1->SG->Cells[Form1->SG->ColCount-1][x]=FloatToStr(masbootstrep[x]);
        }
        if(flag)
        {
            Form1->SG->ColCount+=1;
            ddecode=new DateConvert(Form1->SG,count+1,Form1->top,Form1->SG->ColCount-2,Form1->
>SG->Cells[Form1->right][Form1->top]);

            for(int x=Form1->top;x<count;x++)
                Form1->SG->Cells[Form1->SG->ColCount-1][x]=ddecode->Getdate(x);
            delete ddecode;
        }
        Form1->bottom=count-1;
        Form1->left=Form1->SG->ColCount-1;
        Form1->right=Form1->SG->ColCount-1;
    }
    else
    {
        Form1->SG->ColCount+=1;

        dconvert=new DateConvert(Form1->SG, Form1->bottom+1,Form1->top,Form1->left);
        for(int x=0;x<=Form1->bottom-Form1->top+1;x++)
            mas[x]=dconvert->Getstep(x);
        delete dconvert;
        for(int x=0;x<fiks;x++)
            masbootstrep[x]=mas[x];
        for(int x=count-fiks;x<count;x++)
            masbootstrep[x]=mas[x-(count-fiks)+Form1->bottom+1-fiks];
    }
}

```

```

for(int x=Form1->top+fiks;x<count-fiks;x++)
{
    int index=Form1->top+rand()/32767.0*(Form1->bottom-Form1->top);
    masbootstrep[x]=mas[index];
}
for(int x=Form1->top;x<count;x++)
{
    Form1->SG->Cells[Form1->SG->ColCount-1][x]=FloatToStr(masbootstrep[x]);
}

Form1->SG->ColCount+=1;
ddecode=new DateConvert(Form1->SG,count+1,Form1->top,Form1->SG->ColCount-2,Form1->SG-
>Cells[Form1->left][Form1->top]);
for(int x=Form1->top;x<count;x++)
    Form1->SG->Cells[Form1->SG->ColCount-1][x]=ddecode->Getdate(x);
delete ddecode;

for(int x=Form1->top;x<=Form1->bottom;x++)
    mas[x-Form1->top]=StrToFloat(Form1->SG->Cells[Form1->right][x]);
for(int x=0;x<fiks;x++)
    masbootstrep[x]=mas[x];
for(int x=count-fiks;x<count;x++)
    masbootstrep[x]=mas[x-(count-fiks)+Form1->bottom+1-fiks];
for(int x=Form1->top+fiks;x<count-fiks;x++)
{
    int index=Form1->top+rand()/32767.0*(Form1->bottom-Form1->top);
    masbootstrep[x]=mas[index];
}
Form1->SG->ColCount+=1;
for(int x=Form1->top;x<count;x++)
{
    Form1->SG->Cells[Form1->SG->ColCount-1][x]=FloatToStr(masbootstrep[x]);
}
Form1->bottom=count-1;
Form1->left=Form1->SG->ColCount-2;
Form1->right=Form1->SG->ColCount-1;
}
delete mas;
delete masbootstrep;
buildfts();
}
//-----
MultiStep *calculateforecast(float *mas, int size,int forepoint,float kappa)
{
    MultiStep* forecast=new MultiStep[size-2];
    for(int x=0;x<size-2;x++)
    {
        if(x>=size-forepoint-1)
            mas[x+2]=(forecast[x-1].Getforecast());
        forecast[x].Calculate(mas[x],mas[x+1],mas[x+2],kappa);
    }
    return forecast;
}

```

ftschart.h

```

//-----

#ifndef ftschartH
#define ftschartH
//-----
#include <System.Classes.hpp>
#include <Vcl.Controls.hpp>
#include <Vcl.StdCtrls.hpp>
#include <Vcl.Forms.hpp>
#include <Vcl.ExtCtrls.hpp>
#include <Vcl.Menus.hpp>
#include <VCLTee.Chart.hpp>
#include <VclTee.TeeGDIPlus.hpp>
#include <VCLTee.TeEngine.hpp>
#include <VCLTee.TeeProcs.hpp>
#include <VCLTee.Series.hpp>
//-----
class TForm3 : public TForm
{
__published:        // IDE-managed Components
    TMainMenu *MainMenu1;
    TMenuItem *N1;

```

```

TMenuItem *N2;
TMenuItem *N3;
TMenuItem *N4;
TChart *Chart1;
TMenuItem *view;
TMenuItem *value;
TMenuItem *valuetime;
TMenuItem *separabel;
void __fastcall N2Click(TObject *Sender);
void __fastcall N4Click(TObject *Sender);
void __fastcall valueClick(TObject *Sender);
void __fastcall valuetimeClick(TObject *Sender);
void __fastcall separabelClick(TObject *Sender);
private: // User declarations
public: // User declarations
int posseries[8];
void analizseries();
void clearchart();
    __fastcall TForm3(TComponent* Owner);
};
//-----
extern PACKAGE TForm3 *Form3;
//-----
#endif

```

ftschart.cpp

```

//-----

#include <vcl.h>
#pragma hdrstop

#include "ftschart.h"
#include "main.h"
#include "editftschart.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm3 *Form3;

//-----
__fastcall TForm3::TForm3(TComponent* Owner)
: TForm(Owner)
{
}
//-----
void __fastcall TForm3::N2Click(TObject *Sender)
{
Form4->Visible=true;
}
//-----
void TForm3::clearchart()
{
    Form3->Chart1->SeriesList->Clear();
    Form3->Chart1->View3D=false;
}

void __fastcall TForm3::N4Click(TObject *Sender)
{
    clearchart();
    view->Enabled=false;
}
//-----

void __fastcall TForm3::valueClick(TObject *Sender)
{
    value->Checked=true;
    valuetime->Checked=false;
    separabel->Checked=false;
    Form4->namechart->Text="BP значения";
    Form1->top=posseries[0];
    Form1->bottom=posseries[1];
    Form1->left=posseries[3];
    Form1->right=posseries[3];
    clearchart();
    Form4->addchart();
    Form4->namechart->Text="BP модель значения";
    Form1->top=posseries[4];
}

```

```

Form1->bottom=posseries[5];
Form1->left=posseries[7];
Form1->right=posseries[7];
Form4->addchart();
}
//-----

void __fastcall TForm3::valuetimeClick(TObject *Sender)
{
    value->Checked=false;
    valuetime->Checked=true;
    separabel->Checked=false;
    Form4->namechart->Text="BP(t) значения";
    Form1->top=posseries[0];
    Form1->bottom=posseries[1];
    Form1->left=posseries[2];
    Form1->right=posseries[2];
    clearchart();
    Form4->addchart();
    Form4->namechart->Text="BP(t) модель значения";
    Form1->top=posseries[4];
    Form1->bottom=posseries[5];
    Form1->left=posseries[6];
    Form1->right=posseries[6];
    Form4->addchart();
}
//-----

void __fastcall TForm3::separabelClick(TObject *Sender)
{
    value->Checked=false;
    valuetime->Checked=false;
    separabel->Checked=true;
    Form4->namechart->Text="BP";
    Form1->top=posseries[0];
    Form1->bottom=posseries[1];
    Form1->left=posseries[2];
    Form1->right=posseries[3];
    clearchart();
    Form4->addchart();
    Form4->namechart->Text="BP модель";
    Form1->top=posseries[4];
    Form1->bottom=posseries[5];
    Form1->left=posseries[6];
    Form1->right=posseries[7];
    Form4->addchart();
}
//-----

```

editftschart.h

```

//-----

#ifndef editftschartH
#define editftschartH
//-----
#include <System.Classes.hpp>
#include <Vcl.Controls.hpp>
#include <Vcl.StdCtrls.hpp>
#include <Vcl.Forms.hpp>
//-----
class TForm4 : public TForm
{
__published:        // IDE-managed Components
    TButton *Button1;
    TEdit *namechart;
    TLabel *Label1;
    void __fastcall Button1Click(TObject *Sender);
private:            // User declarations
public:             // User declarations
    void addchart();
    __fastcall TForm4(TComponent* Owner);
};
//-----
extern PACKAGE TForm4 *Form4;
//-----
#endif

```

15
1116130. 01165-01 12 01
editftschart.cpp

```
//-----
#include <vcl.h>
#pragma hdrstop

#include "editftschart.h"
#include "ftschart.h"
#include "main.h"
#include "DateConvert.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm4 *Form4;

//-----
__fastcall TForm4::TForm4(TComponent* Owner)
: TForm(Owner)
{
}
//-----
void __fastcall TForm4::Button1Click(TObject *Sender)
{
    Form1->left=Form1->SG->Selection.Left;
    Form1->right=Form1->SG->Selection.Right;
    Form1->top=Form1->SG->Selection.Top;
    Form1->bottom=Form1->SG->Selection.Bottom;
    addchart();
}
//-----
void TForm4::addchart()
{
    int i=1;
    TChartSeries *ch=new TLineSeries(Form3->Chart1);
    if(Form1->right==Form1->left)
    {
        Form3->Chart1->BottomAxis->Automatic=true;
        Form3->Chart1->TopAxis->Automatic=true;

        if(Form1->SG->Cells[Form1->right][Form1->top].Pos(".")==0)
        {
            for(int x=Form1->top;x<=Form1->bottom;x++)
            {
                if(Form1->SG->Cells[Form1->right][x]!="")
                    ch->AddXY(i,StrToFloat(Form1->SG->Cells[Form1->right][x]));
                i++;
            }
        }
        else
        {
            float *mas=new float[Form1->bottom-Form1->top+1];
            DateConvert *dconvert;
            dconvert=new DateConvert(Form1->SG, Form1->bottom,Form1->top,Form1->left);
            float temp;
            for(int x=0;x<=Form1->bottom-Form1->top;x++)
            {
                mas[x]=dconvert->Getstep(x);
                temp=mas[x];
            }
            for(int x=0;x<=Form1->bottom-Form1->top;x++)
            {
                ch->AddXY(i,mas[x]);
                i++;
            }

            delete dconvert;
            delete mas;
        }
        Form3->Chart1->BottomAxis->Automatic=true;
        Form3->Chart1->TopAxis->Automatic=true;
        ch->Title=Form4->namechart->Text;
        Form3->Chart1->AddSeries(ch);
    }
    else
    {

```

1116130. 01165-01 12 01

```

TDate *ndate;
Form3->Chart1->BottomAxis->Automatic=false;
Form3->Chart1->TopAxis->Automatic=false;
if(Form1->SG->Cells[Form1->left][Form1->top].Pos(".")!=0)
{
    ndate=new TDate(Form1->SG->Cells[Form1->left][Form1->bottom]);
    Form3->Chart1->BottomAxis->Maximum=ndate->Val;
    delete ndate;
    ndate=new TDate(Form1->SG->Cells[Form1->left][Form1->top]);
    Form3->Chart1->BottomAxis->Minimum=ndate->Val;
    delete ndate;
}
ch->XValues->DateTime=true;
float min,max;
min=StrToFloat(Form1->SG->Cells[Form1->right][Form1->top]);
max=StrToFloat(Form1->SG->Cells[Form1->right][Form1->top]);
for(int x=Form1->top;x<=Form1->bottom;x++)
{
    if(Form1->SG->Cells[Form1->left][x].Pos(".")!=0)
    if(Form1->SG->Cells[Form1->right][x].Pos(".")==0)
    if(Form1->SG->Cells[Form1->right][x]!="")
    if(Form1->SG->Cells[Form1->left][x]!="")
    {
        ndate=new TDate(Form1->SG->Cells[Form1->left][x]);
        if(max<StrToFloat(Form1->SG->Cells[Form1->right][x]))
            max=StrToFloat(Form1->SG->Cells[Form1->right][x]);
        if(min>StrToFloat(Form1->SG->Cells[Form1->right][x]))
            min=StrToFloat(Form1->SG->Cells[Form1->right][x]);

        ch->AddXY(ndate->Val,StrToFloat(Form1->SG->Cells[Form1->right][x]));
        delete ndate;
    }
}
Form3->Chart1->TopAxis->Maximum=max;
Form3->Chart1->TopAxis->Minimum=min;
ch->Title=Form4->namechart->Text;
Form3->Chart1->AddSeries(ch);
}
}

```

analizseries.h

```

//-----
#ifndef analizseriesH
#define analizseriesH
//-----
#include <System.Classes.hpp>
#include <Vcl.Controls.hpp>
#include <Vcl.StdCtrls.hpp>
#include <Vcl.Forms.hpp>
//-----
class TForm5 : public TForm
{
__published:        // IDE-managed Components
    TEdit *count;
    TLabel *Label1;
    TButton *Button1;
    void __fastcall Button1Click(TObject *Sender);
private:             // User declarations
public:              // User declarations
    __fastcall TForm5(TComponent* Owner);
};
//-----
extern PACKAGE TForm5 *Form5;
//-----
#endif

```

analizseries.cpp

```

//-----
#include <vcl.h>
#pragma hdrstop

#include "analizseries.h"
#include "main.h"
#include "DateConvert.h"

```



```
#include "ftsms.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm5 *Form5;
//-----
__fastcall TForm5::TForm5(TComponent* Owner)
: TForm(Owner)
{
}
//-----
void __fastcall TForm5::Button1Click(TObject *Sender)
{
    Form2->analizseries(StrToInt(count->Text));
    Form2->buildfts();
}
//-----
```

exelload.h

```
//-----
#ifndef exelloadH
#define exelloadH
//-----
#include <System.Classes.hpp>
#include <Vcl.Controls.hpp>
#include <Vcl.StdCtrls.hpp>
#include <Vcl.Forms.hpp>
//-----
class TForm6 : public TForm
{
__published:        // IDE-managed Components
    TButton *Button1;
    TButton *Button2;
    void __fastcall Button2Click(TObject *Sender);
    void __fastcall Button1Click(TObject *Sender);
private:            // User declarations
public:              // User declarations
    __fastcall TForm6(TComponent* Owner);
};
//-----
extern PACKAGE TForm6 *Form6;
//-----
#endif
```

exelload.cpp

```
//-----
#include <vcl.h>
#pragma hdrstop
#include "exelload.h"
#include "main.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm6 *Form6;
//-----
__fastcall TForm6::TForm6(TComponent* Owner)
: TForm(Owner)
{
}
//-----
void __fastcall TForm6::Button2Click(TObject *Sender)
{
    Form1->app.OlePropertyGet("WorkBooks",1).OleProcedure("Close");
    Form6->Close();
}
//-----
void __fastcall TForm6::Button1Click(TObject *Sender)
{
    Form1->app.OlePropertyGet("Cells").OlePropertyGet("SpecialCells",
    xlCellType::xlCellTypeLastCell).OleProcedure("Activate");
    int rows = Form1->app.OlePropertyGet("ActiveCell").OlePropertyGet("Row");
    int cols = Form1->app.OlePropertyGet("ActiveCell").OlePropertyGet("Column");
    Form1->SG->RowCount=rows;
    Form1->SG->ColCount=cols;
    for(int x=0;x<rows;x++)
    for(int y=0;y<cols;y++)
    {
        Form1->SG->Cells[y][x]=Form1->
        >app.OlePropertyGet("WorkBooks",1).OlePropertyGet("WorkSheets",1).OlePropertyGet("Cells",x+1,y+1).OlePropertyGet("Value");
    }
}
```

```

    }
}
//-----
bootstrep.h
//-----
#ifndef bootstrepH
#define bootstrepH
//-----
#include <System.Classes.hpp>
#include <Vcl.Controls.hpp>
#include <Vcl.StdCtrls.hpp>
#include <Vcl.Forms.hpp>
//-----
class TForm7 : public TForm
{
__published:      // IDE-managed Components
    TButton *Button1;
    TEdit *countseries;
    TLabel *Label3;
    TLabel *Label5;
    TEdit *fiks;
    void __fastcall Button1Click(TObject *Sender);
private:          // User declarations
public:           // User declarations
    __fastcall TForm7(TComponent* Owner);
};
//-----
extern PACKAGE TForm7 *Form7;
//-----
#endif

```

```

bootstrep.cpp
//-----
#include <vcl.h>
#pragma hdrstop
#include "bootstrep.h"
#include "ftsms.h"
#include "main.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm7 *Form7;
//-----
__fastcall TForm7::TForm7(TComponent* Owner)
: TForm(Owner)
{
}
//-----
void __fastcall TForm7::Button1Click(TObject *Sender)
{
    Form1->left=Form1->SG->Selection.Left;
    Form1->right=Form1->SG->Selection.Right;
    Form1->top=Form1->SG->Selection.Top;
    Form1->bottom=Form1->SG->Selection.Bottom;
    bool flag=true;
    int x,y;
    y=Form1->left;
    x=Form1->right;
    x=x-y;
    if(x>1)
    {
        ShowMessage("Обрано більше двох рядів");
        flag=false;
    }
    y=Form1->top;
    x=Form1->bottom;
    x=x-y;
    if(fiks->Text=="")
    {
        ShowMessage("Пусте поле фіксовані точки");
        flag=false;
    }
    if(StrToInt(fiks->Text)<0)
    {
        ShowMessage("Поле фіксовані точки не може бути від'ємним");
        flag=false;
    }
    if(flag)

```

```

        if(x<StrToInt(fiks->Text)*2)
        {
            ShowMessage("Фіксованих точок більше вхідного ряду");
            flag=false;
        }
        if(flag)
        if(StrToInt(fiks->Text)*2<=StrToInt(countseries->Text))
            Form2->bootstrep(StrToInt(fiks->Text),StrToInt(countseries->Text));
        else
            ShowMessage("Розмір нового НЧР не може бути менши чім за "+IntToStr(StrToInt(fiks-
>Text)*2));
    }
    //-----

```

MultiStep.h

```

//-----
#ifndef MultiStepH
#include<vcl.h>
#define MultiStepH
class MultiStep
{
private:
    float kappa;
    float forecast;

public:
    MultiStep();
    Calculate(float x1,float x2,float x3, float kappa);
    float Getforecast();
};
//-----
#endif

```

MultiStep.cpp

```

//-----
#pragma hdrstop
#include "MultiStep.h"
//-----
#pragma package(smart_init)
MultiStep::MultiStep()
{
}
//-----
MultiStep::Calculate(float x1,float x2,float x3,float kappa)
{
    this->kappa=kappa;
    float *datepoint=new float[3];
    datepoint[0]=x1;
    datepoint[1]=x2;
    datepoint[2]=x3;
    float delta=(datepoint[2]-datepoint[1])-(datepoint[1]-datepoint[0]);
    float deltad=delta*0.95;
    float deltau=delta*1.05;
    if(deltad<0)
        if(this->kappa<=(deltad))
            if(this->kappa>=(deltau))
                this->forecast=((datepoint[1]*(0.5)+datepoint[0]+datepoint[2]*(0.5))/2);
            else
                this->forecast=((datepoint[1]*(0.75)+datepoint[0]+datepoint[2]*(0.25))/2);
        else
            this->forecast=((datepoint[1]*(0.25)+datepoint[0]+datepoint[2]*(0.75))/2);
    else
        if(this->kappa>=(deltad))
            if(this->kappa<=(deltau))
                this->forecast=(datepoint[1]*(0.5)+datepoint[0]+datepoint[2]*(0.5))/2;
            else
                this->forecast=(datepoint[1]*(0.25)+datepoint[0]+datepoint[2]*(0.75))/2;
        else
            this->forecast=(datepoint[1]*(0.75)+datepoint[0]+datepoint[2]*(0.25))/2;
    delete datepoint;
}
//-----
float MultiStep::Getforecast()
{
    return this->forecast;
}
//-----

```

DateConvert.h

```

//-----

```

```

#ifndef DateConvertH
#include "main.h"
#include <DateUtils.hpp>
#define DateConvertH
class DateConvert
{
private:
    TDate *mdate;
    int *mstep;
    int size;

public:
    DateConvert(TStringGrid *SG,int Bottom,int Top,int left);
    DateConvert(TStringGrid *SG,int Bottom,int Top,int index,UnicodeString start);
    int Getstep(int index);
    UnicodeString Getdate(int index);
};
//-----
#endif

DateConvert.cpp

//-----
#pragma hdrstop
#include "DateConvert.h"
DateConvert::DateConvert(TStringGrid *SG,int Bottom,int Top,int index)
{
    size=Bottom-Top;
    mdate=new TDate [size+1];
    mstep=new int[size+1];
    float temp;
    UnicodeString str;
    for(int x=Top;x<=Bottom;x++)
        if(SG->Cells[index][x]!="")
        {
            str=SG->Cells[index][x];
            TDate *ndate=new TDate(SG->Cells[index][x]);
            mdate[x-Top]=*ndate;
        }
    mstep[0]=0;
    for(int x=1;x<=size;x++)
    {
        mstep[x]=DaysBetween(mdate[x-1],mdate[x]);
        temp= mstep[x];
    }
}
DateConvert::DateConvert(TStringGrid *SG,int Bottom,int Top,int index,UnicodeString start)
{
    size=Bottom-Top;
    mdate=new TDate [size];
    mstep=new int[size];
    for(int x=Top;x<Bottom;x++)
        if(SG->Cells[index][x]!="")
        {
            mstep[x-Top]=StrToFloat(SG->Cells[index][x]);
        }

    TDate *ndate=new TDate(start);
    mdate[0]=IncDay(*ndate,mstep[0]);
    for(int x=1;x<size;x++)
    {
        mdate[x]=IncDay(mdate[x-1],mstep[x]);
    }
}
int DateConvert::Getstep(int index)
{
    if(index<=size)
        return mstep[index];
    return 0;
}
UnicodeString DateConvert::Getdate(int index)
{
    if(index<size)
        return mdate[index].DateString();
    return "-";
}
//-----
#pragma package(smart_init)

```

**Дослідження процедур моделювання та прогнозування дискретних
антиперсистентних процесів**

Скалозуб В. В., Клименко І.В., Шаповал Д. О.

Український державний університет науки і технологій, Україна

Комп'ютерне моделювання натеper являється одним із основних засобів аналізу та прогнозування складних процесів у багатьох сферах діяльності. Моделі, методи та засоби щодо коректного та достовірного застосування процедур комп'ютерне моделювання були і залишаються актуальними. Складність процесів дослідження являється головним із факторів, які визначають можливості та ефективність процедур аналізу та прогнозування сучасних систем, с тому числі процесів функціонування залізничного транспорту.

При аналізі та моделюванні складних систем в багатьох випадках в якості вихідних даних використовують часові послідовності процесів, які представляють показники досліджуваних величин. Формування та інтерпретація таких послідовностей даних потребує в тому числі коректного застосування сукупності математико-статистичних методів аналізу. Також ці методи вимагають досить значну кількість даних, виконання умов певних припущень, наприклад, постулати нормальності ін. Серед важливих вимог до формування та застосування методів комп'ютерного моделювання визначається наявність трендів досліджуваних процесів. Розвиток методів та засобів хаотичної динаміки дозволив встановити вимоги, які по-суті забезпечують можливість застосування процедур моделювання та прогнозування процесів, представлених часовими послідовностями, часовими рядами (ЧР). Таким показником являється параметр Херста, за яким виконується класифікація процесів на персистентні (трендостійкі), антиперсистентні (для них передбачається лише можливість зміни тенденції розвитку), хаотичні, При цьому натеper методи аналізу, комп'ютерного моделювання та прогнозування антиперсистентні часових послідовностей (АЧП) досліджені не в повній мірі.

При моделюванні АЧП даних, які представляють складні процеси різної природи, виникають питання щодо визначення можливості коректного формування комп'ютерних моделей, процедур визначення параметрів станів процесів, методики дослідження властивостей часових послідовностей, точність отриманої моделі тощо.

У доповіді приведено результатів досліджень щодо можливостей та ефективності алгоритмів, призначених для класифікації, моделювання та прогнозування показників антиперсистентних процесів. Представлені розглянуті структури процедур аналізу та класифікації антиперсистентних процесів, проведено аналіз напрямків щодо удосконалення алгоритмів класифікації та короткострокового прогнозування антиперсистентних часових послідовностей (АЧП), запропонована методика та процедури дослідження АЧР шляхом порівняльного аналізу чисельної ефективності та точності алгоритмів класифікації та інтерполяції даних моделей АЧП. На основі зазначеної методики класифікації та прогнозування АЧР проведені дослідження функціональної ефективності алгоритмів моделювання. В роботі були визначені та представлені особливості структури процесів моделювання антиперсистентних числових послідовностей. При порівнянні алгоритмів класифікації АЧП було встановлено, що найбільше детальною та сталою являється класифікація на основі схем агрегування даних АЧП без перетину у послідовностях елементів що узагальнюються, тобто рівнів часових рядів. Отримані результати були використані при формування програмного забезпечення для дослідження моделей дискретних антиперсистентних процесів. Практичне значення мають результати щодо засобів комп'ютерного моделювання АЧП, розробки та порівняльного аналізу алгоритмів моделювання, результати порівняльного аналізу чисельної ефективності алгоритмів класифікації та процедури моделювання антиперсистентних числових послідовностей.