МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інтелектуальних інформаційних систем

_____Юрій КОНДРАТЕНКО

«____»_____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА

ІНТЕЛЕКТУАЛЬНА СИСТЕМА МОДЕЛЮВАННЯ ТА ПРОГНОЗУВАННЯ НА ОСНОВІ МЕТОДІВ КОМБІНУВАННЯ

Спеціальність 122 Комп'ютерні науки

Освітня програма «Інтелектуальні інформаційні системи»

Здобувач _____ Максим МЕЛЬНИЧУК

«___»_____ 2024 р.

Керівник д-р техн. наук, доцент _____ Ірина КАЛІНІНА

«___»_____ 2024 р.

Миколаїв – 2024

Чорноморський національний університет імені Петра Могили
(повне найменування закладу вищої освіти)

Факультет	Комп'ютерних наук
Кафедра	Інтелектуальних інформаційних систем
Рівень вищої освіти	Другий (магістерський)
Освітній ступень	Магістр
Спеціальність	122 Комп'ютерні науки
Освітня програма	Інтелектуальні інформаційні системи

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних інформаційних систем

_____ Юрій КОНДРАТЕНКО

«____» _____ 2024 р.

ЗАВДАННЯ
на кваліфікаційну роботу здобувача

_____Мельничука Максима Сергійовича_____
((прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: «Інтелектуальна система моделювання і прогнозування на основі методів комбінування».

Керівник роботи: Калініна Ірина Олександрівна, в. о. професора кафедри ІІС д-р техн. наук, доцент.

Затверджена наказом ЧНУ ім. Петра Могили від «03» червня 2024 р. № 140/1.

2. Строк представлення кваліфікаційної роботи «16» грудня 2024 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні: система прогнозування попиту на електроенергію в енергомережі України; часовий ряд попиту на електроенергію в енергомережі України.

4. Перелік питань, що підлягають розробці: огляд існуючих методів прогнозування часових рядів і комбінування прогнозів, аналіз і попередня обробка часового ряду,

створення прогнозних моделей на для прогнозування часового ряду, комбінування прогнозів і порівняння точності окремих і комбінованих моделей.

5. Перелік графічних матеріалів: презентація.

Керівник роботи _____ Ірина КАЛІНІНА
(Особистий підпис) *(Власне ім'я ПРІЗВИЩЕ)*

Здобувач _____ Максим МЕЛЬНИЧУК
(Особистий підпис) *(Власне ім'я ПРІЗВИЩЕ)*

Дата видачі завдання «07» червня 2024 р.

КАЛЕНДАРНИЙ ПЛАН
кваліфікаційної роботи

Тема: Інтелектуальна система моделювання та прогнозування на основі методів комбінування

№	Найменування роботи	Початок	Закінчення	Примітки
1	Отримання завдання на виконання КР	03.06.2024	07.06.2024	виконано
2	Аналіз предметної області та постановка задачі	10.06.2024	20.06.2024	виконано
3	Огляд літературних джерел за темою кваліфікаційної роботи, зокрема аналіз публікацій щодо використання методів комбінування прогнозів для підвищення точності прогнозу	21.06.2024	01.07.2024	виконано
4	Огляд існуючих методів прогнозування часових рядів та комбінування прогнозів, аналіз та попередня обробка часового ряду	01.09.2024	25.10.2024	виконано
5	Створення окремих прогнозних моделей та комбінування прогнозів, оцінка точності прогнозів	26.10.2024	21.11.2024	виконано
6	Перший попередній захист КР на засіданні комісії кафедри	22.11.2024	22.11.2024	виконано
7	Корегування роботи за результатами попереднього захисту	23.11.2024	05.12.2024	виконано
8	Другий попередній захист КР на засіданні комісії кафедри	06.12.2024	06.12.2024	виконано
9	Доробка та остаточне оформлення КР	07.12.2024	10.02.2024	виконано
10	Подання КР, її електронної копії та інших документів (відгуку, рецензії) до захисту	16.12.2024	17.12.2024	виконано

Керівник роботи _____ Ірина КАЛІНІНА
(Особистий підпис) *(Власне ім'я ПРІЗВИЩЕ)*

Здобувач _____ Максим МЕЛЬНИЧУК
(Особистий підпис) *(Власне ім'я ПРІЗВИЩЕ)*

Дата складання календарного плану
«19» червня 2024 р.

АНОТАЦІЯ

до кваліфікаційної роботи
здобувача групи 601м ЧНУ ім. Петра Могили

Мельничука Максима Сергійовича

на тему: **“ ІНТЕЛЕКТУАЛЬНА СИСТЕМА МОДЕЛЮВАННЯ І ПРОГНОЗУВАННЯ НА ОСНОВІ МЕТОДІВ КОМБІНУВАННЯ ”**

Актуальність роботи: комбінування прогнозів отриманих за допомогою різних прогнозних моделей дозволяє отримати точніший прогноз зміни значень часового ряду, що може бути застосовано в багатьох галузях.

Об'єкт дослідження – процеси та інформаційні технології машинного навчання й прогнозування.

Предмет дослідження – методи прогнозування часових рядів та комбінування прогнозів.

Мета дослідження: покращення точності прогнозування значень часового ряду із застосуванням методів комбінування прогнозів.

Пояснювальна записка до магістрської кваліфікаційної роботи складається зі вступу, чотирьох розділів, висновків та додатків.

У першому розділі розглядаються існуючі методи і підходи до прогнозування часових рядів і комбінування прогнозів.

У другому розділі виконується аналіз публікацій на тему комбінування прогнозів і описується структура інформаційної системи.

У третьому розділі описується виконання аналізу і попередньої обробки набору даних.

У четвертому розділі описується моделювання і оцінка розроблених окремих і комбінованих прогнозних моделей.

Магістрська кваліфікаційна робота містить 87 сторінок, 52 рисунки, 5 таблиць, 41 використане джерело та 3 додатки.

Ключові слова: прогнозування, часовий ряд, комбінування прогнозів.

ABSTRACT

to the qualification work by the student of the group 601m of Petro Mohyla Black Sea National University

Melnychuk Maksym

“ INTELLIGENT MODELING AND FORECASTING SYSTEM BASED ON COMBINATION METHODS ”

Relevance of work: combining different forecast models allows to get a more accurate forecast of time series, which can be applied in many industries.

The object of work is time series forecasting.

The subject of work is methods of time series forecasting and forecast combination.

The purpose of work is improving time series forecast accuracy using forecast combination methods.

Explanatory note to the bachelor’s qualification work contains introduction, four sections, conclusions and appendices.

In the first section, the methods and approaches for time series forecasting are described.

In the second, the publications about forecast combinations are analyzed and structure of informational system is described.

In the third section describes the analysis and preprocessing of dataset.

The last section describes modeling and accuracy estimation of the individual and combined forecasts.

Master’s qualification work contains 87 pages, 52 pictures, 5 tables, 41 references and 3 appendices.

Keywords: forecasting, time series, forecast combination.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ОЕС – об’єднана енергетична система;
ARIMA – autoregressive integrated moving average;
ENTSO-E – European Network of Transmission System Operators for Electricity;
GAM – generalized additive model;
GLM – generalized linear model;
KPSS – тест Квятковського-Філліпса-Шмідта-Шина;
LOCF – last observation carried forward;
MAE – mean absolute error;
MAPE – mean absolute percentage error;
MSE – mean squared error;
RMSE – root mean squared error.

ВСТУП

Сучасний світ характеризується складністю і динамічністю процесів, що відбуваються в різних сферах життєдіяльності людини. В таких умовах ефективні методи прогнозування набувають важливого значення в різних галузях, зокрема для підвищення ефективності будь-якої економічної діяльності.

Електроенергетика є прикладом галузі, в якій точні прогнози можуть сприяти ухваленню більш оптимальних рішень, зокрема прогнозування попиту на електроенергію в енергосистемі держави. Нажаль у зв'язку з атаками на об'єкти енергетичної інфраструктури України з боку російської федерації наша держава періодично стикається з нестачею електроенергії, і прогнозування попиту на електроенергію стає ще більш актуальним, наприклад для визначення оптимального періоду для ремонту енергетичних об'єктів або визначення обсягів закупівлі електроенергії в разі її очікуваної нестачі.

При прогнозуванні економічних показників, зокрема попиту, ми маємо справу з деякою кількістю спостережуваних значень показника у різні моменти часу. Аналізуючи динаміку зміни минулих значень показника в часі передбачається побудувати прогноз його майбутніх значень. Таким чином ми маємо справу з задачею прогнозування часового ряду.

Поширені у задачі прогнозування часових рядів методи машинного навчання, такі як узагальнена адитивна модель, модель експоненційного згладжування і модель ARIMA, дозволяють отримати доволі точний прогноз із врахуванням сезонних закономірностей, що вочевидь мають великий вплив на попит на електроенергію. Класичним підходом є побудова декількох прогнозних моделей на навчальній вибірці і перевірка їх точності на тестовій вибірці з подальшим вибором найточнішої моделі. Проте різні моделі можуть давати доволі різні прогнози, і вибір серед отриманих окремих прогнозів найоптимальнішого прогнозу, тобто з найкращою точністю на навчальній і тестовій вибірках, не завжди є оптимальним рішенням, адже призводить до ймовірної втрати інформації про закономірності, які

не відображає обрана модель, проте можуть відображати інші, дещо менш точні на тестовій вибірці, моделі.

Підходом, який дозволяє вирішити згадану проблему і отримати більш точний прогноз за побудовані окремими моделями є комбінування прогнозів. Різні методи комбінування прогнозів передбачають отримання із множини декількох окремих прогнозів одного комбінованого прогнозу, який за даними багатьох досліджень виявляється точнішим за найкращі окремі прогнози.

Метою цієї роботи є покращення точності прогнозування попиту на електроенергію із застосуванням методів комбінування прогнозів.

Для досягнення мети необхідно проаналізувати існуючі методи прогнозування часових рядів та комбінування прогнозів, виконати аналіз та попередню обробку обраного для прогнозування часового ряду, побудувати декілька окремих прогнозних моделей з використанням різних методів машинного навчання, оцінити їх точність та порівняти її з точністю комбінованих прогнозів, отриманих з використанням різних методів комбінування прогнозів.

1 ОГЛЯД ІСНУЮЧИХ МЕТОДІВ І ПІДХОДІВ ДО ПРОГНОЗУВАННЯ І КОМБІНУВАННЯ ПРОГНОЗІВ

1.1 Задача прогнозування часових рядів

Часовий ряд – це ряд точок даних, перелічених в хронологічному порядку, взятих, як-правило, через рівні проміжки часу. Будь-які дані, зібрані шляхом послідовних спостережень протягом часу, є часовим рядом. Кожне окреме значення в часовому ряді називають спостереженням або рівнем часового ряду, йому у відповідність ставиться певний момент часу або номер за порядком. Часовий ряд суттєво відрізняється від простої вибірки даних, оскільки при його аналізі враховується взаємозв’язок змін з часом, а не лише статистичні характеристики вибірки.

Задача прогнозування часових рядів полягає в тому, щоб ґрунтуючись на попередніх спостереженнях значень певної змінної, та інших даних, доступних на момент виконання прогнозу, спрогнозувати як змінюватиметься значення змінної в подальшому. Часто значення факторів, що впливають на прогнозоване значення, також не відомі заздалегідь, тому зазвичай методи прогнозування часових рядів ґрунтуються тільки на інформації про значення прогнозованої змінної, не намагаючись виявити фактори, що впливають на її поведінку. Натомість такі моделі аналізують динаміку зміни значень ряду, виявляють трендові і сезонні закономірності [1-4].

Задача прогнозування, зокрема у застосуванні до часових рядів, зазвичай складається з п’яти основних етапів, наведених у табл. 1.1.

Таблиця 1.1 – Основні етапи задачі прогнозування

Етап	Короткий опис
Визначення задачі	Визначення задачі вимагає розуміння того, як і ким будуть використовуватись прогнози, яким чином збираються і зберігаються дані для виконання прогнозування. У застосуванні до часових рядів тут слід визначити частоту, з якою фіксуються спостереження і з якою робляться прогнози (наприклад щогодинні, щоденні, щомісячні дані), горизонт прогнозування – на який час вперед виконується прогноз.
Збір даних та інформації	Методи прогнозування, застосовні до заданої задачі, багато в чому залежать від наявних даних. Кількісне прогнозування може виконуватись якщо доступна достатня кількість історичних даних, і розумно припустити, що деякі аспекти минулих закономірностей будуть продовжуватись в майбутньому. Якщо достатньої кількості даних нема в наявності, може застосовуватись прогнозування на основі суджень. Іноді не всі дані будуть однаково корисними для побудови моделей, наприклад старі дані можуть бути менш корисними, ніж новіші.
Попередній (розвідувальний) аналіз	На цьому етапі зібрані дані аналізуються, зокрема будуються графіки даних, виявляються закономірності, наприклад тренд, сезонні закономірності, циклічні закономірності, наявність в даних відхилень, тощо.

Кінець таблиці 1.1

Етап	Короткий опис
Вибір і підгонка моделей	На цьому етапі за наявними даними будуються прогнозні моделі. Кожна модель є сама по собі штучною конструкцією на основі набору допущень, і зазвичай включає в себе один або декілька параметрів, що мають бути оцінені. Зазвичай будують і потім порівнюють кілька потенційних моделей.
Оцінка прогнозної моделі та її використання	Коли модель обрана і її параметри оцінено, модель використовують для отримання прогнозів. Результативність моделі може бути оцінено належним чином тільки в тому разі, якщо доступні дані за прогнозований період. Існує ряд методів, що дозволяють оцінити точність прогнозу.

На рис. 1.1 схематично зображено етапи побудови прогнозної моделі.

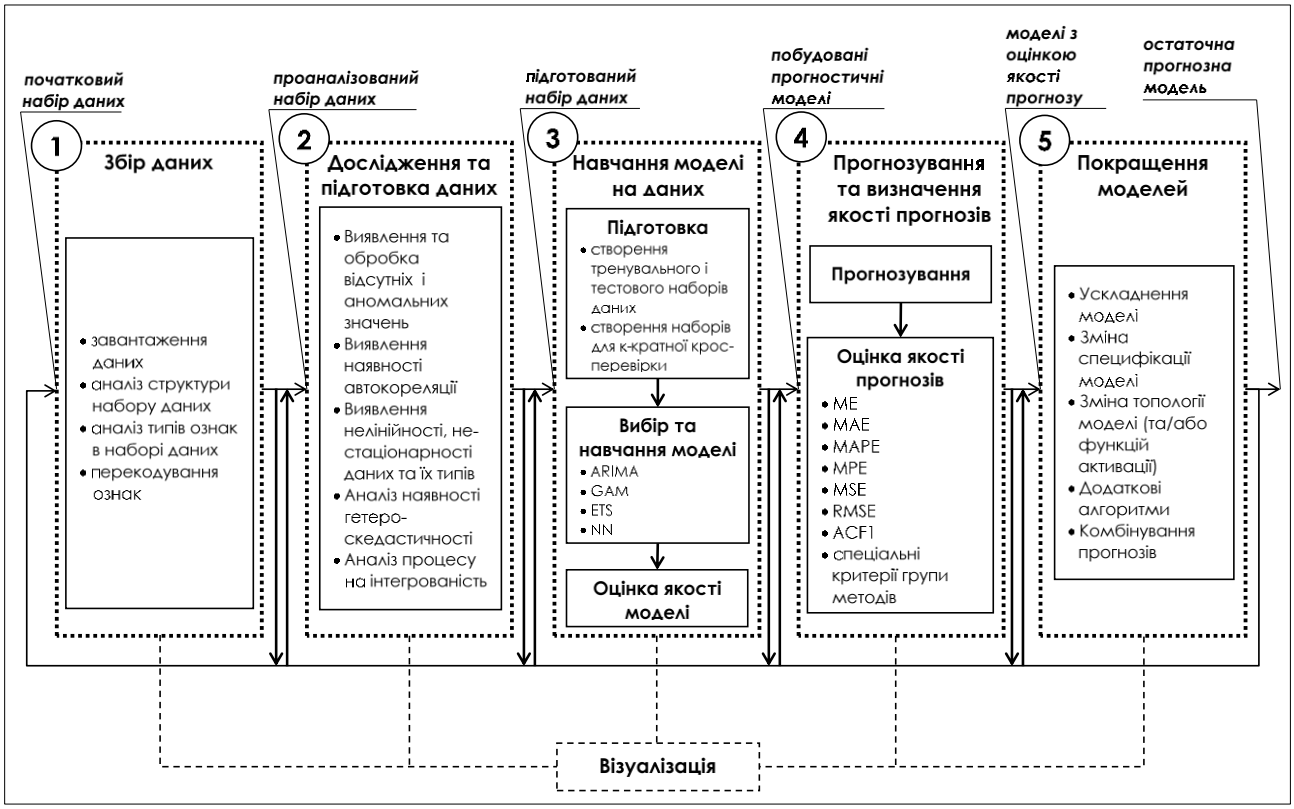

Рисунок 1.1 – Схема побудови прогнозної моделі

Кожен етап побудови прогнозної моделі передбачає можливість повернення на попередні етапи в разі потреби і супроводжується візуалізацією результатів – для часового ряду перш за все графіками реальних і прогнозованих значень ряду.

1.2 Огляд окремих моделей прогнозування

1.2.1 Узагальнена адитивна модель

Моделі лінійної регресії є досить простими для сприйняття моделями для прогнозування значення залежної змінної на основі лінійної залежності між прогнозованою змінної і змінними-предикторами. Рівняння лінійної регресії з кількома змінними-предикторами і однією залежною змінною наведене нижче.

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \cdots + \beta_n x_{n,t} + \varepsilon_t \quad (1.1)$$

де y_t – значення прогнозованої змінної в момент часу t;

$x_{1,t} \ldots x_{n,t}$ – значення змінних-предикторів в момент часу t;

$\beta_0 \ldots \beta_n$ – коефіцієнти лінійної залежності, значення яких треба оцінити для побудови моделі;

n – кількість змінних-предикторів;

ε_t – випадкова похибка, яка пояснює відхилення спостережень від лінійної залежності.

Значення коефіцієнтів перед змінними-предикторами $\beta_0 \ldots \beta_n$ зазвичай оцінюються методом найменших квадратів, їх називають коефіцієнтами лінійної регресії.

Розширенням моделей лінійної регресії для випадку багатьох залежних змінних є модель загальної лінійної регресії, відповідно множинна лінійна регресія вигляду (1.1) є окремим випадком загальної лінійної регресійної моделі.

Очевидним недоліком лінійних моделей є те, що реальні дані часто не відповідають лінійній залежності. Одним зі способів подолання цього недоліку є введення функції зв'язку, що пов'язуватиме лінійну комбінацію змінних-предикторів із залежною змінною, що має нелінійний характер. В такому разі модель набуває вигляду (1.2), така модель називається узагальненою лінійною моделлю (generalized linear model – GLM).

$$g(y_t) = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \cdots + \beta_n x_{n,t} + \varepsilon_t \quad (1.2)$$

де y_t – значення прогнозованої змінної в момент часу t;

$x_{1,t}$ ... $x_{n,t}$ – значення змінних-предикторів в момент часу t;

$\beta_0 \dots \beta_n$ – коефіцієнти лінійної залежності, значення яких треба оцінити для побудови моделі;

n – кількість змінних-предикторів;

ε_t – випадкова похибка, яка пояснює відхилення спостережень від лінійної залежності;

$g(y_t)$ – функція зв'язку.

Іншим можливим способом моделювання нелінійної залежності є лінійна комбінація згладжуючих функцій, що приймають змінні-предиктори в якості аргументів (1.3). Така модель називається адитивною.

$$y_t = \beta_0 + f_1(x_{1,t}) + f_2(x_{2,t}) + \cdots + f_n(x_{n,t}) + \varepsilon_t \quad (1.3)$$

де y_t – значення прогнозованої змінної в момент часу t;

$x_{1,t}$ ... $x_{n,t}$ – значення змінних-предикторів в момент часу t;

n – кількість змінних-предикторів;

ε_t – випадкова похибка, яка пояснює відхилення спостережень від лінійної залежності;

$f_i(\ )$ – згладжуюча функція для і-ї змінної-предиктора.

Поєднанням двох описаних вище підходів узагальненої лінійної моделі та адитивної моделі є узагальнена адитивна модель (generalized additive model – GAM), тобто її можна описати залежністю (1.4).

$$g(y_t) = \beta_0 + f_1(x_{1,t}) + f_2(x_{2,t}) + \cdots + f_n(x_{n,t}) + \varepsilon_t \quad (1.4)$$

Серед переваг узагальненої адитивної моделі виділяють можливість моделювати різноманітні нелінійні взаємозв'язки і можливість інтерпретації результатів, що полегшує розуміння зв'язку між предиктором і залежною змінною, серед недоліків – обчислювальну складність, необхідність великої вибірки для навчання моделі, складність підбору параметрів моделі [5,6].

Нещодавно була запропонована модель Prophet, доступна в пакеті fable.prophet. Модель на основі підходу узагальнених регресійних моделей була розроблена компанією Facebook для прогнозування даних зі тижневою і річною сезонністю і врахуванням ефекту свят і впливових подій. Загалом модель описується залежністю (1.5).

$$y(t) = g(t) + s(t) + h(t) + e_t \quad (1.5)$$

де $y(t)$ – прогнозована змінна;

$g(t)$ – функція, що апроксимує тренд часового ряду;

$s(t)$ – функція, що апроксимує сезонні коливання;

$h(t)$ – функція, що відображає ефекти свят і інших впливових подій;

e_t– нормально розподілена випадкова похибка.

Для апроксимації тренду використовується шматочно-лінійна регресія з автоматично відібраними або явно вказаними вузловими точками, сезонна компонента складається з членів Фур'є відповідних періодів, святкові ефекти, такі

як державні свята, спортивні або культурні події, тощо, враховуються в вигляді простих індикаторних змінних. Модель Prophet найкраще підходить для прогнозування часових рядів, що мають сильну сезонність і кілька сезонів історичних даних [1-3,7].

1.2.2 Експоненційне згладжування

Одним із поширених методів прогнозування часових рядів є метод експоненційного згладжування. Особливістю методів експоненційного згладжування є прогнозування на основі попередніх спостережень, ваги яких експоненційно зменшуються із застарінням спостережень. Концепція експоненційного зважування виходить з припущення, що недавні спостереження є більш суттєвими для прогнозування майбутніх значень, ніж більш віддалені у часі.

Найпростішим варіантом експоненційного згладжування є метод простого експоненційного згладжування, що добре підходить для прогнозування даних без чітко вираженої трендової або сезонної закономірності. Прогнози в моделі експоненційного згладжування розраховуються з використанням середньозважених значень попередніх спостережень, ваги яких експоненційно зменшуються з віддаленням у минуле (1.6).

$$y_{T+1|T} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots \qquad (1.6)$$

де y_i – значення прогнозованої змінної в момент часу і;

α – параметр згладжування.

Швидкість, з якою ваги спостережень зменшуються з віддаленням у минуле регулюється параметром α, більше його значення відповідає більшій швидкості спадання, тобто вплив недавніх спостережень буде більшим, а давніших – меншим. На рис. 1.2 наведено три графіки зміни коефіцієнтів спостережень з часом для

різних значень α. Для достатньо великого розміру вибірки сума ваг спостережень буде приблизно рівною одиниці при будь-якому значенні α.

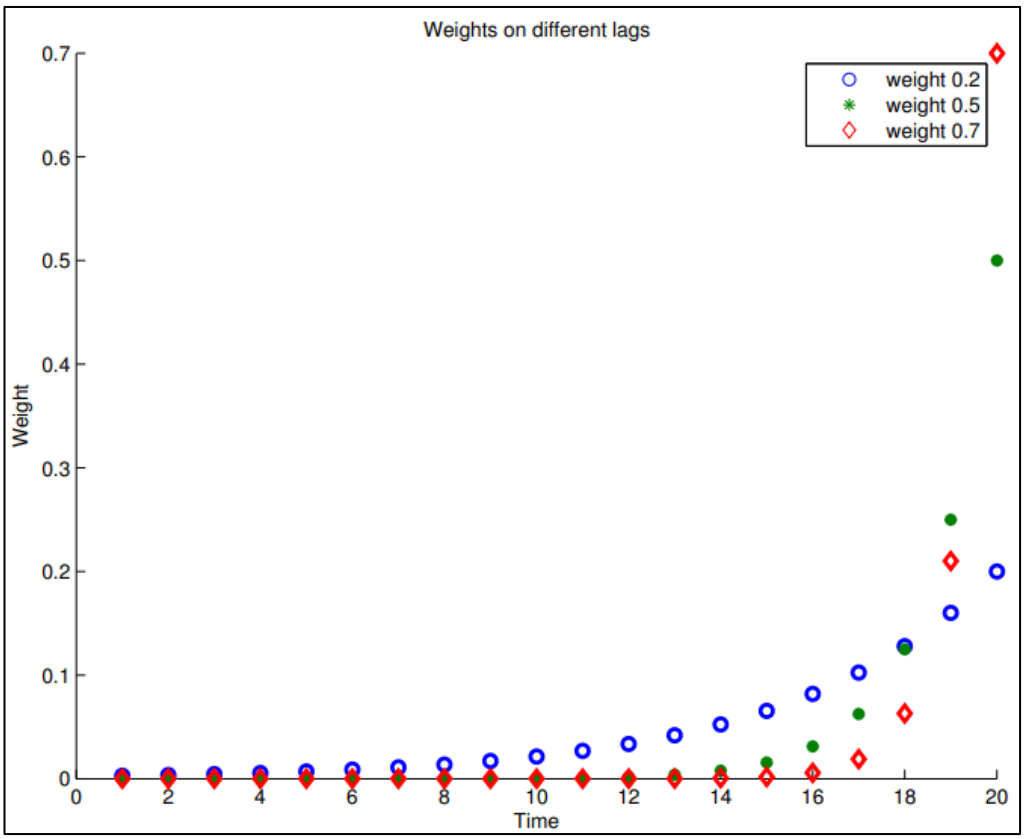

Рисунок 1.2 – Графіки значень ваг при різних коефіцієнтах α

Рівняння (1.6) можна виразити у компонентній формі (1.7), яка для простого експоненціального згладжування міститиме тільки одну компоненту – рівень l_t.

$$\hat{y}_{t+h|t} = l_t, \qquad (1.7)$$
$$l_t = \alpha y_t + (1-\alpha) l_{t-1}$$

де $\hat{y}_{t+h|t}$ – прогнозоване значення змінної y в момент часу t+h , що ґрунтується на даних $y_1 \ldots y_t$;

α – параметр експоненційного згладжування.

Більш складні моделі експоненційного згладжування здатні також прогнозувати дані з трендом і сезонністю. Наведемо приклад методу що враховує тренд - метод лінійного тренду Хольта. Цей метод має дві компоненти: компоненту рівня і компоненту тренду (1.8). Як бачимо, компонента тренду теж має власний параметр згладжування.

$$\hat{y}_{t+h|t} = l_t + hb_t, \tag{1.8}$$
$$l_t = \alpha y_t + (1-\alpha)(l_{t-1} + b_{t-1})$$
$$b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1}$$

де l_t – компонента рівня;

b_t– компонента тренду;

s_t – сезонна компонента;

α – параметр згладжування для компоненти рівня;

β – параметр згладжування для компоненти тренду.

Оскільки на практиці метод лінійного тренду часто завищує прогноз, існує також метод демпфованого тренду (1.9).

$$\hat{y}_{t+h|t} = l_t + (\phi + \phi^2 + \cdots + \phi^h)b_t \tag{1.9}$$
$$l_t = \alpha y_t + (1-\alpha)(l_{t-1} + \phi b_{t-1})$$
$$b_t = \beta(l_t - l_{t-1}) + (1-\beta)\phi b_{t-1}$$

де l_t – компонента рівня;

b_t– компонента тренду;

s_t – сезонна компонента;

α – параметр згладжування для компоненти рівня;

β – параметр згладжування для компоненти тренду;

ϕ – параметр демпфування, $0 < \phi < 1$.

При застосуванні методу демпфованого тренду короткострокові прогнози будуть близькі до методу лінійного тренду, а довгострокові – наближатимуться до постійного значення.

Окрім тренду моделі експоненційного згладжування також можуть враховувати сезонність. Для прикладу наведемо методи Хольта-Вінтерса з адитивною і мультиплікативною сезонністю – формули (1.10) і (1.11) відповідно.

$$\hat{y}_{t+h|t} = l_t + hb_t + s_{t+h+m(k+1)}, \tag{1.10}$$
$$l_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1}),$$
$$b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1},$$
$$s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}$$

$$\hat{y}_{t+h|t} = (l_t + hb_t) * s_{t+h+m(k+1)}, \tag{1.11}$$
$$l_t = \alpha\left(\frac{y_t}{s_{t-m}}\right) + (1-\alpha)(l_{t-1} + b_{t-1}),$$
$$b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1},$$
$$s_t = \gamma\left(\frac{y_t}{l_{t-1} - b_{t-1}}\right) + (1-\gamma)s_{t-m}$$

де l_t – компонента рівня;

b_t– компонента тренду;

s_t – сезонна компонента;

α – параметр згладжування для компоненти рівня;

β – параметр згладжування для компоненти тренду;

γ – параметр згладжування для сезонної компоненти;

m – кількість спостережень в одному сезонному періоді;

k – цілочисельна частина (h-1)/m, потрібна щоб оцінки сезонних індексів, що використовуються для прогнозування, бралися з останнього року вибірки;

Таким чином залежно від наявності і типів компонент існує дев’ять методів експоненційного згладжування, які позначаються комбінацією із двох літер, як наведено в табл. 1.2.

Таблиця 1.2 – Класифікація методів експоненційного згладжування

Трендова компонента	Сезонна компонента		
	Відсутня (N)	Адитивна (A)	Мультиплікативна (M)
Відсутня (N)	(N,N)	(N,A)	(N,M)
Адитивна (A)	(A,N)	(A,A)	(A,M)
Адитивна демпфована (A_d)	(A_d,N)	(A_d,A)	(A_d,M)

Кожен метод має свої випадки, в яких його застосування буде більш оптимальним. Наприклад адитивна сезонність підходить для випадків, коли сезонні коливання залишаються приблизно постійними протягом всього ряду, тоді як мультиплікативна сезонність підходить якщо сезонні коливання змінюються пропорційно рівню ряду [1-4,8,9].

1.2.3 Моделі ARIMA

Іншою поширеною моделлю прогнозування часових рядів є модель авторегресійного інтегрованого ковзного середнього (autoregressive integrated moving average – ARIMA). Модель ARIMA є комбінацією авторегресії, диференціювання і ковзного середнього.

Авторегресійні моделі – це моделі, що в якості змінних-предикторів використовують попередні значення прогнозованої змінної. Авторегресійну модель порядку p (скорочено записують AR(p)) можна виразити рівнянням (1.12).

$$y_t = \mathrm{c} + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t \quad (1.12)$$

де y_t– значення прогнозованої змінної в момент часу t;

c – константа;

$\phi_1 \dots \phi_n$ – коефіцієнти лінійної залежності;

ε_t – білий шум.

Як бачимо, авторегресійна модель подібна до множинної лінійної регресії з минулими значеннями прогнозованої змінної в якості предикторів.

Моделі ковзного середнього мають дещо подібний підхід, проте замість використання минулих значень прогнозованої змінної в регресії модель ковзного середнього використовує минулі прогрозні помилки в регресійно-подібній моделі (1.13).

$$y_t = \mathrm{c} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} \quad (1.13)$$

де y_t– значення прогнозованої змінної в момент часу t;

c – константа;

$\theta_1 \dots \theta_n$ – коефіцієнти лінійної залежності;

ε_t – білий шум.

Таку модель називають моделлю ковзного середнього порядку q – MA(q).

Авторегресійна модель і модель ковзного середнього, так само як модель ARIMA, що їх використовує, передбачають, що дані часового ряду, з яким ми працюємо, є стаціонарними. Стаціонарність – це властивість часового ряду зберігати свої статистичні властивості незмінними в часі. Таким чином часові ряди, що мають тренд або сезонність не є стаціонарними. Стаціонарний часовий ряд не матиме передбачуваних закономірностей в довгостроковій перспективі.

Стаціонарність часового ряду можна виявити за відсутністю автокореляції (кореляції значень часового ряду зі значеннями цього ж ряду зсунутими у часі), а також за спеціальними тестами, такими як тест Квятковського-Філліпса-Шмідта-Шина (KPSS).

Один зі способів отримання стаціонарного часового ряду з нестаціонарного це обчислити різниці між почерговими спостереженнями. Ця процедура називається диференціюванням часового ряду. Диференціювання часового ряду не обов'язково приводить до стаціонарного часового ряду – в такому випадку можна виконати диференціювання повторно – диференціювання другого порядку. На практиці майже ніколи нема необхідності виходити за рамки диференціювання другого порядку.

Якщо поєднати авторегресію, ковзне середнє і диференціювання, то отримаємо несезонну модель ARIMA(p,d,q), в якій:

- p – порядок авторегресійної частини;
- d – порядок диференціювання;
- q – порядок частини з ковзним середнім.

Несезонна модель ARIMA може бути відображена у вигляді залежності (1.14).

$$y_t = \mathrm{c} + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t \qquad (1.14)$$

де y_t – значення часового ряду в момент часу t, можливо неодноразово диференційовані;

Сезонна модель ARIMA містить додаткову сезонну частину з аналогічними трьома компонентами, що позначаються $(P,D,Q)_m$, де m – кількість спостережень в одному сезонному періоді.

Приблизно оцінити значення p і q для моделей вигляду ARIMA(0,d,q) або ARIMA(p,d,0) можна за допомогою графіків функцій автокореляції (ACF) і

часткової автокореляції (PACF). Графік функції автокореляції показує кореляцію між y_t та y_{t-k} для різних значень k, тобто кореляцію значень часового ряду з власними значеннями зсунутими на k періодів часу (лагами). Проте y_t та y_{t-k} можуть корелювати не через зв'язок між собою, а через те що обидві змінні корелюють зі змінними $y_{t-1} \ldots y_{t-k+1}$. Функція часткової автокореляції усуває цей ефект.

Дані можуть відповідати моделі ARIMA(p,d,0) якщо графіки ACF і PACF диференційованих даних демонструють закономірності: ACF є експоненційно затухаючим або синусоїдальним, спостерігається значний сплеск у лазі p на PACF. Дані можуть відповідати моделі ARIMA(0,d,q) якщо графіки ACF і PACF диференційованих даних демонструють закономірності: PACF є експоненційно затухаючим або синусоїдальним, спостерігається значний сплеск в лазі q на ACF.

Для оцінки параметрів моделі в програмних пакетах, що реалізують метод ARIMA, використовуються спеціальні функції. Порядок диференціювання d може бути оцінено повторним застосуванням згаданого тесту KPSS або інших тестів на стаціонарність, значення p і q підбираються шляхом мінімізації значення інформаційних критеріїв, наприклад критерію Акаіке (AIC), скоригованого критерію Акаіке (AICc), баєсівського критерію (BIC). Значення коефіцієнтів $\theta_1 \ldots \theta_n$ та $\phi_1 \ldots \phi_n$ оцінюють за методом найменших квадратів [1-3,6,10-12].

1.2.4 Нейромережеві моделі

Штучна нейронна мережа – це математична модель, побудована за принципом організації та функціонування біологічних нейронних мереж живих організмів. Нейромережеві моделі прогнозування допускають складні нелінійні зв'язки між залежною змінною і змінними-предикторами, що робить доцільним їх використання зокрема для прогнозування часових рядів.

Нейронна мережа може розглядатися як мережа штучних нейронів, організованих шарами. Кожен штучний нейрон має кілька входів та один вихід. Загальну схему будови штучного нейрона наведено на рис. 1.3.

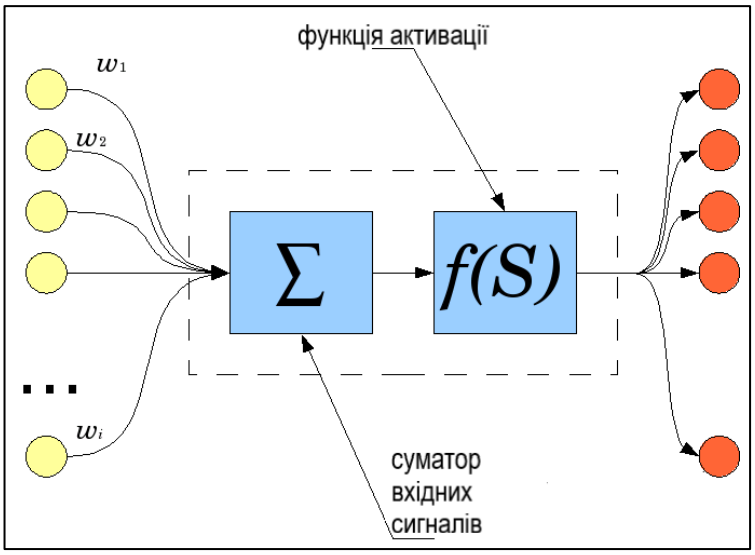

Рисунок 1.3 – Схема будови штучного нейрона

Сигнали, що подаються на входи множаться на вагові коефіцієнти, сумуються, і передаються до активаційної функції, результат якої подається на вихід нейрона. Сигнали на виходах нейронів першого шару нейромережі подаються на входи нейронів наступних шарів. Нейронна мережа містить вхідний шар, на який подаються значення змінних-предикторів, вихідний шар, де отримується результат, і може містити кілька прихованих шарів. На рис. 1.4 наведено схему нейронної мережі з 4 входами, 1 виходом і одним прихованим шаром з трьома нейронами.

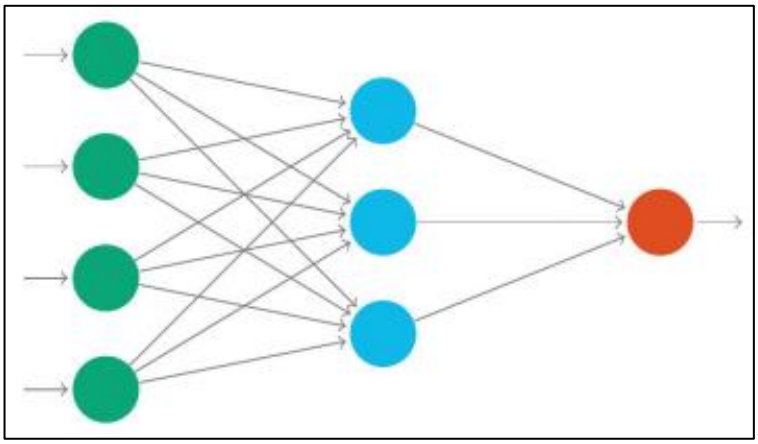

Рисунок 1.4 – Структура нейронної мережі з одним прихованим шаром нейронів

Навчання нейронної мережі полягає у підборі таких значень вагових коефіцієнтів на входах штучних нейронів, що забезпечували б правильну роботу моделі. Навчання нейронної мережі відбувається з застосуванням навчального набору даних, що містить множину вхідних значень і відповідних до них вихідних значень.

Підходом, що дозволяє застосувати нейронні мережі для прогнозування часових рядів є нейромережева авторегресія. В цьому методі зсунуті в часі значення часового ряду використовуються в якості змінних-предикторів, подібно до того як ще відбувається в авторегресійній моделі. В даному випадку попередні спостереження подаються на входи нейронної мережі. Модель NNAR(p,k) матиме p входів і k нейронів в одному прихованому шарі, відповідно для прогнозування значення y_t на входи подаватимуться значення $y_{t-1}…y_{t-p}$. Для прогнозування на один крок на входи моделі подаються відповідні значення. Прогнозування більш ніж на один крок виконується ітеративно: спочатку здійснюється прогноз на один крок, потім це значення разом з попередніми спостереженнями використовується для прогнозування наступного значення, і так до обчислення усіх необхідних прогнозів. Ця модель може бути модифікована для часових рядів з сезонністю, адже в такому випадку корисно враховувати останні спостереження за той самий сезон, що й прогнозоване значення. Така модель позначається $NNAR(p,P,k)_m$ і має входи $y_{t-1}, y_{t-2}, \dots, y_{t-p}, y_{t-m}, y_{t-2m}, \dots, y_{t-Pm}$ і k нейронів в прихованому шарі, m – кількість спостережень в одному сезонному періоді. [1-3,13-15].

1.3 Аналіз поширених методів комбінування прогнозів

1.3.1 Загальний огляд комбінування прогнозів

Як правило будуючи кілька прогнозних моделей на основі різних методів або одного методу з різними параметрами для одного й того ж часового ряду серед них

обирають один найбільш оптимальний метод. Це традиційний підхід, який виходить з припущення, що найкращий метод існує і може бути виявлений [16].

Оцінку моделей для порівняння їх ефективності виконують на тестових даних, що не використовувались для оцінки параметрів моделі, а отже можуть відображати ефективність прогнозної моделі в застосуванні до нових даних. Для оцінки, як правило, підсумовують помилки моделі, тобто відхилення прогнозованого моделлю значення від реального, певним чином, отримуючи прогнозні помилки моделі: середню абсолютну помилку (mean absolute error, MAE), корінь середньої квадратичної помилки (root mean squared error, RMSE), середню абсолютну відсоткову помилку (mean absolute percentage error, MAPE), тощо [1-3]. Найоптимальнішою вважається модель, що демонструє найменші значення помилок на тестових даних.

Описаний підхід дозволяє вибрати оптимальний метод, проте він пов’язаний з ризиком того, що обрана модель виявиться не найкращою, і демонструватиме гірші результати на інших даних [17]. Вибір тільки однієї моделі із множини успішних моделей може призвести до втрати цінної інформації, наявної в альтернативних моделях [18].

Альтернативним підходом є комбінування прогнозів – об’єднання інформації, наявної в окремих прогнозах для отримання більш якісного прогнозу. Існують різні підходи до комбінування прогнозів, які демонструють гарні результати у покращенні точності прогнозів [18]. Ефективність комбінування прогнозів у порівнянні з вибором найкращого окремого прогнозу показано зокрема у статтях [16,17,19,20].

1.3.2 Методи усереднення прогнозів

Найпростішим методом комбінування прогнозів є їх просте усереднення, що можна відобразити формулою (1.15).

$$C_t = \frac{\sum_{p=1}^{P} y_{pt}}{P} \quad (1.15)$$

де C_t – комбінований прогноз в момент часу t;

P – кількість окремих прогнозних моделей, що комбінуються;

y_{pt} – значення прогнозу отримане окремою моделлю p для моменту часу t.

Попри простоту цей метод може давати не гірші результати, ніж інші, складніші методи комбінування прогнозів. В літературі цю ситуацію називають загадкою комбінування прогнозів (forecast combination puzzle)[21-24] або загадкою рівних вагових коефіцієнтів (equal weights puzzle) [25]. Це може пояснюватися тим, що оцінка вагових коефіцієнтів за певним методом збільшує дисперсію прогнозу, відповідно він демонструє більші значення середньої квадратичної помилки [26,27], цей ефект є більш суттєвим за комбінування великої кількості прогнозів, коли вплив кожного окремого прогнозу на результат не надто великий [27].

Ефективність комбінування прогнозів шляхом усереднення залежить від дисперсії похибок окремих прогнозних моделей. Наприклад якщо маємо дві прогнозні моделі з прогнозами y_{1t} та y_{2t} , то похибка комбінованого прогнозу обчислюватиметься наступним чином:

$$e_{Ct} = y_t - y_{Ct} = y_t - \frac{y_{1t}+y_{2t}}{2} = \frac{e_{1t}+e_{2t}}{2} \quad (1.16)$$

де y_t – справжнє значення прогнозованої змінної в момент часу t;

y_{Ct}– комбінований прогноз в момент часу t;

y_{1t} та y_{2t} – значення прогнозу отримані окремими моделями;

e_{Ct} – похибка комбінованого прогнозу

e_{1t} , e_{2t} – похибки окремих прогнозів, $e_{pt} = y_t - y_{pt}$

Оскільки окремі прогнози є неупередженими, що має забезпечувати коректно побудована прогнозна модель, то і комбінований прогноз є неупередженим, це означає що похибки прогнозів мають середнє значення 0, тобто математичне сподівання $E[e_{pt}] = 0$. Дисперсія похибки комбінованого прогнозу в такому разі розраховується наступним чином:

$$Var[e_{Ct}] = Var\left[\frac{e_{1t}+e_{2t}}{2}\right] = E\left[\left(\frac{e_{1t}+e_{2t}}{2}\right)^2\right] = \frac{1}{4}E[e_{1t}^2 + 2e_{1t}e_{2t} + e_{2t}^2] = \quad (1.17)$$

$$= \frac{1}{4}(E[e_{1t}^2] + 2E[e_{1t}e_{2t}] + E[e_{2t}^2]) = \frac{1}{4}\sigma_1^2 + 2\frac{E[e_{1t}e_{2t}]}{\sigma_1\sigma_2}\sigma_1\sigma_2 + \sigma_1^2 =$$

$$= \frac{\sigma_1^2+2\rho\sigma_1\sigma_2+\sigma_2^2}{4}$$

де $\sigma_p^2 = Var[e_{pt}]$ – дисперсія похибок р-ї прогнозної моделі;

$\sigma_c^2 = Var[e_{ct}]$ – дисперсія похибок комбінованого прогнозу;

e_{Ct} – похибка комбінованого прогнозу

e_{1t} , e_{2t} – похибки окремих прогнозів, $e_{pt} = y_t - y_{pt}$

ρ – коефіцієнт кореляції між e_{1t} і e_{2t}

Вважаючи похибки окремих прогнозів незалежними одна від одної можна спростити формулу (1.16):

$$\sigma_c^2 = \frac{\sigma_1^2+\sigma_2^2}{4} \quad (1.18)$$

Таким чином дисперсія похибок комбінованого прогнозу є значно меншою за дисперсію окремих прогнозів, наприклад якщо $\sigma_1^2 = \sigma_2^2 = 100$, то $\sigma_c^2 = 50$. Але навіть за високого ступеня кореляції між похибками окремих прогнозів дисперсія похибок комбінованого прогнозу буде меншою, наприклад для тих самих значень σ_1^2 і σ_2^2 і якщо взяти $\rho = 0{,}8$ отримаємо $\sigma_c^2 = 90$. Тут слід зауважити, що ситуація змінюється, якщо дисперсії похибок окремих прогнозів сильно різняться між

собою. Наприклад якщо $\sigma_1^2 = 100$, $\sigma_2^2 = 16$, $\rho = 0$, то $\sigma_c^2 = 29$, тобто дисперсія похибок другого окремого прогнозу буде меншою, за дисперсію похибок комбінованого прогнозу. Таким чином просте усереднення може бути ефективно застосоване в випадку, коли дисперсії похибок окремих прогнозів мають значення, що не сильно відрізняються.

Поряд із простим усередненням існують дещо складніші методи усереднення прогнозів, покликані виправити його недоліки, одним з яких є те, що на результат простого усереднення великий вплив мають викиди. Більш стійким до викидів підходом є взяття медіани (1.19) [24,28].

$$C_t = median(y_{pt}) \tag{1.19}$$

де C_t – комбінований прогноз в момент часу t;

y_{pt} – окремий прогноз в момент часу t.

Проміжним підходом між медіаною і простим усередненням є обрізане середнє (trimmed mean). Метод полягає в тому, що середнє значення враховується не з усіх прогнозованих значень, а з ближчих до середнього, відсікаючи найбільші і найменші λ% значень, де λ це параметр обрізки [24,29]. Значення комбінованого прогнозу для випадку обрізаного середнього може бути обчислене за формулою (1.20).

$$C_t = \frac{1}{P(1-2\lambda)} \sum_{i=\lambda P+1}^{(1-\lambda)P} y_{ip} \tag{1.20}$$

де C_t – комбінований прогноз в момент часу t;

y_{pt} – окремий прогноз в момент часу t;

P – кількість окремих прогнозів;

λ – параметр обрізки.

Ще одним подібним методом є вінзоризоване усереднення. Вінзоризація – це метод перетворення даних шляхом обмеження їхніх екстримальних значень, названий на честь Чарльза Вінзора, американського математика та інженера який запропонував цей метод. Він полягає у скиданні екстримальних значень до вказаного процентиля, що є ефективним засобом обробки викидів, який при цьому не викидає їх з аналізу цілком, що може бути його перевагою перед обрізаним середнім [29].

1.3.3 Метод мінімальної дисперсії

Метод мінімальної дисперсії, вперше запропонований у важливій для теми комбінації прогнозів статті [30], полягає у обчисленні комбінованих прогнозних значень як зваженої суми значень окремих прогнозів (1.21), де вагові коефіцієнти окремих прогнозів обчислюються відповідно до точності прогнозу, тобто вони обернено пропорційні помилкам прогнозів (1.22) [19,20].

$$C_t = \sum_{p=1}^{P} w_p y_{pt} \qquad \sum_{p=1}^{P} w_p = 1 \tag{1.21}$$

де C_t – комбінований прогноз в момент часу t;

y_{pt} – окремий прогноз в момент часу t;

w_p – ваговий коефіцієнт окремої прогнозної моделі p;

$$w_p = \frac{(\sum_{i=1}^{N}(y_i - \hat{y}_{pi})^2)^{-1}}{(\sum_{p=1}^{P}(\sum_{i=1}^{N}(y_i - \hat{y}_{pi})^2)^{-1})} \tag{1.22}$$

де w_p – ваговий коефіцієнт окремої прогнозної моделі p;

y_i – реальне значення i-го спостереження;

$\hat{y}_{pi}$ – прогнозоване окремою моделлю p значення i-го спостереження;

N – кількість спостережень у тестовій вибірці;

P – кількість окремих прогнозних моделей.

Для створення більшого розриву між вагами, щоб більш вдалі моделі мали суттєвішу перевагу, можна використати коефіцієнт k зі значенням більшим за одиницю (1.23) [28].

$$w_p = \frac{(\sum_{i=1}^{N}(y_i - \hat{y}_{pi})^2)^{-k}}{(\sum_{p=1}^{P}(\sum_{i=1}^{N}(y_i - \hat{y}_{pi})^2)^{-k})} \quad (1.23)$$

Подібним за суттю підходом є метод оберненого рангу. Прогнози ранжуються у порядку зростання помилки прогнозу, після чого ваги обчислюються за оберненими значеннями рангів прогнозів (1.24).

$$w_p = \frac{Rank_p{}^{-1}}{\sum_{p=1}^{P} Rank_p{}^{-1}} \quad (1.24)$$

де w_p – ваговий коефіцієнт окремої прогнозної моделі p;

$Rank_p$ – ранг прогнозу;

P – кількість окремих прогнозних моделей.

Підхід оберненого рангу вважається більш стійким до викидів [18].

1.3.4 Метод лінійної регресії

Ще одним підходом до комбінування прогнозів є використання є метод регресії, що застосовує лінійну регресійну модель, в якій в якості прогнозованої змінної виступає значення комбінованого прогнозу, а змінними-предикторами є значення окремих прогнозів (1.25) [18,19,31].

$$C_t = \alpha_0 + \sum_{p=1}^{P} \alpha_p y_{pt} \quad (1.25)$$

де C_t – комбінований прогноз в момент часу t;

y_{pt} – окремий прогноз в момент часу t;

$\alpha_0 \dots \alpha_P$ – коефіцієнти лінійної регресії, які треба оцінити.

Значення коефіцієнтів лінійної регресії $\alpha_0 \dots \alpha_P$ оцінюються методом найменших квадратів або найменшого абсолютного відхилення [18,19,31]. В випадку якщо кількість прогнозних моделей велика порівняно з розміром навчальної вибірки результат може погіршуватись, тому для відбору оптимальної підмножини моделей можуть бути застосовані критерії відбору, такі як критерій Акаіке (AIC) і баєсівський критерій (BIC) [31].

1.4 Огляд програмних засобів створення та комбінування прогнозних моделей доступних в R

Часовий ряд можна розглядати як список чисел (спостережень) разом з інформацією про те коли ці числа були записані (індекси спостережень) – зручним засобом для збереження таких даних і роботи з ними в R є об’єкти tsibble. Об’єкт tsibble дозволяє зберігати один або декілька часових рядів із зазначенням часових міток. Можна використовувати спеціальні функції для модифікації часового ряду, наприклад select() для вибору певних стовпців, filter() для вибору певних рядків що задовольняють умову, summarize() для підсумування даних, mutate() для створення нових змінних [1,3,34].

Пакет R forecast надає методи та інструменти для відображення та аналізу прогнозів однофакторних часових рядів. Деякі функції пакету наведені в табл. 1.2 [1,3,35].

Таблиця 1.3 – Деякі функції пакету forecast

Назва	Опис
meanf	Погноз за методом простого усереднення спостережень
ets	Підгонка моделі експоненційного згладжування до часового ряду. Параметри моделі, такі як значення параметрів згладжування, наявність компонент та їх тип, можуть бути задані користувачем або будуть підігнані автоматично.
holt	Підгонка моделі Хольта (моделі експоненційного згладжування з трендом)
hw	Підгонка моделі Хольта-Вінтерса (моделі експоненційного згладжування з трендом і сезонністю)
Arima	Підгонка моделі ARIMA з вказаними параметрами p,d і q для несезонної частини і P, D і Q для сезонної частини до вказаного часового ряду.
auto.arima	Підгонка моделі ARIMA з автоматичним відбором параметрів моделі до вказаного часового ряду.
nnetar	Підгонка моделі на основі нейронної мережі прямого поширення сигналу з одним прихованим шаром з указаною кількістю входів і нейронів прихованого шару до вказаного часового ряду.

Пакет R ForecastComb надає реалізацію багатьох методів комбінування прогнозів. Опис функцій пакету наведено в табл. 1.2 [36,37].

Таблиця 1.4 – Функції пакету ForecastComb

Назва	Опис
auto_combine	Автоматичний підбір моделі комбінування прогнозів на основі вказаного критерію (RMSE, MAE або MAPE)
comb_BG	Комбінування прогнозів за методом мінімальної дисперсії на основі підходу запропонованого в статті Бейтса і Ґрейнджера [30]
comb_CLS comb_OLS	Комбінування прогнозів з використанням регресійної моделі підігнаної за методом найменших квадратів з обмеженнями (CLS – constrained least squares) або без них (OLS – ordinary least squares). Обмеження полягають у тому, що вагові коефіцієнти прогнозів в сумі повинні дорівнювати одиниці.
comb_CSR	Комбінування прогнозів з побудовою багатьох регресійних моделей, серед яких за інформаційними критеріями обирається найкраща
comb_EIG1 comb_EIG2 comb_EIG3 comb_EIG4	Різні варіанти підходу до комбінування прогнозів на основі власних векторів матриці середніх квадратичних помилок прогнозу.
comb_LAD	Комбінування прогнозів з використанням регресійної моделі підігнаної за методом найменшої абсолютної похибки.
comb_MED	Комбінування прогнозів методом медіани.
comb_SA	Комбінування прогнозів методом простого усереднення.
comb_TA	Комбінування прогнозів методом обрізаного середнього.
comb_WA	Комбінування прогнозів методом вінзоризованого усереднення.

Висновки до розділу 1

Задача прогнозування часових рядів передбачає збір даних, виконання аналізу і попередньої обробки даних, побудови і оцінки декількох прогнозних моделей. Серед методів, що можуть бути застосовані до побудови моделей: узагальнені адитивні моделі, моделі експоненційного згладжування, моделі ARIMA, моделі на основі нейромережевої авторегресії. Ефективним засобом отримання кращого прогнозу з кількох моделей, на противагу вибору оптимальної окремої моделі, є комбінування прогнозів, що може виконуватись як шляхом простого усереднення, так і складнішими методами. Також було розглянуто пакети мови програмування R, що надають засоби аналізу часових рядів, створення прогнозних моделей та їх комбінування.

2 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ПРОГНОЗУВАННЯ НА ОСНОВІ МЕТОДІВ КОМБІНУВАННЯ

2.1 Комбінування прогнозів. Аналіз публікацій

Ефективність комбінування прогнозів різних моделей у порівнянні з відбором найкращої окремої моделі показана у багатьох дослідженнях. Так в дослідженні [19] було порівняно три окремі прогнозні моделі (модель на основі штучної нейронної мережі, модель ARIMA і модель експоненційного згладжування) і три підходи до комбінування прогнозів (просте усереднення, мінімальна дисперсія і лінійна регресія) на 500 симульованих часових рядах по 200 спостережень в кожному. Для оцінювання прогнозів використовували показники MAE, MAPE, RMSE та коефіцієнт U Тейла. В результаті серед окремих моделей модель штучної нейронної мережі виявилась найбільш ефективною, але вона може поступатись комбінованим моделям простого усереднення і мінімальної дисперсії і сильно програє регресійній комбінованій моделі.

Так само до висновку про ефективність комбінованих моделей у порівнянні з підходом відбору найкращої індивідуальної моделі приходять автори статті [17]. Ними взято 3003 набори даних, серед яких є дані взяті щорічно, щоквартально, щомісячно, щоденно і інші, для перевірки гіпотези щодо більшої ефективності комбінування прогнозів у порівнянні з вибором найкращої окремої моделі. Серед методів прогнозування розглядалися різні варіанти експоненційного згладжування (просте експоненційне згладжування, метод Хольта, метод демпфованого тренду), моделі ARIMA, моделі нейромережевої авторегресії. Для комбінування прогнозів використовувалося тільки просте усереднення, проте розглядалися усі можливі комбінації прогнозів. Точність пронозів оцінювали за допомогою симетричної MAPE (sMAPE). В результаті автори статті приходять до висновку, що вибір однієї

окремої прогнозної моделі пов’язаний з більшим ризиком і може мати значно меншу ефективність, ніж комбінування декількох прогнозів.

У дослідженні [20] також показано ефективність комбінованих прогнозів у порівнянні з окремими прогнозними моделями – в якості них було взято модель регресії опорних векторів (support vector regression), модель ARIMA, модель експоненційного згладжування з мультиплікативною сезонністю (метод Вінтерса) і наївні моделі, прогнози яких виходять із припущення, що прогнозоване значення буде збігатися з останнім відомим значенням в той самий сезонний період. Згадані методи було застосовано до прогнозування кількості туристичних поїздок з різними цілями до Великої Британії. Прогнози комбінували з використанням підходів простого усереднення, мінімальної дисперсії і дисконтованої середньої квадратичної помилки. В статті звертається увага що комбінація з використанням підмножини більш оптимальних прогнозних моделей показує кращі результати, ніж комбінація усіх доступних моделей, а оптимальна кількість прогнозів для комбінування складає від двох до п’яти.

У статті [22] розглядається так звана загадка комбінації прогнозів, яка полягає у тому, що ускладнення комбінаційної моделі не призводить до гарантованого покращення точності моделі у порівнянні з простішими методами комбінування. Автори статті застосовують нелінійні методи комбінування прогнозів, засновані на алгоритмах машинного навчання, зокрема на основі дерев рішень і модель FFORMA. Розглядалися дві задачі: комбінування різних вже готових прогнозів значення індексу споживчих цін в Європейському Союзі і прогнозування макроекономічних показників США, де використовувалося кілька окремих моделей, серед яких модель усереднення історичних значень часового ряду, наївні моделі, модель експоненційного згладжування, модель ARIMA, модель нейромережевої авторегресії. У результаті із застосуванням нелінійні методи комбінування прогнозів було отримано кращі результати, ніж із використанням простого усереднення.

Натомість у статті [24], виконували прогнозування різних економічних показників країн, результати порівняння простих моделей усереднення прогнозів або медіани із різними складнішими моделями (дисконтованої середньої квадратичної похибки, обчислення ваг на основі критеріїв AIC і BIC, регресійної моделі з визначенням коефіцієнтів на основі методу найменших квадратів, тощо) виявилися не на користь останніх.

У статті [16] прогнози 10 окремих моделей, серед яких наївна модель, модель ковзного середнього, кілька моделей експоненційного згладжування та модель лінійної регресії, комбінували з використанням методу мінімальної дисперсії. Прогнози виконувалися для багатьох різних часових рядів і з різними прогнозними горизонтами. В результаті було показано, що комбіновані прогнози були точнішими за прогнози окремих моделей в більшості випадків за винятком великих прогнозних горизонтів.

У статті [37] використовували моделі ARIMA, простого експоненційного згладжування, експоненційного згладжування з демпфованим трендом, модель нейромережевої авторегресії і тета-модель, а також їх комбінації різними методами для прогнозування виробництва електроенергії. Точність прогнозів оцінювали за значенням MAE. В результаті точність вищу за точність найкращого окремого прогнозу було отримано при комбінуванні прогнозів методом лінійної регресії із коефіцієнтами оціненими методом найменших квадратів. У висновку автори зазначають, що комбінування прогнозів дозволяє значно зменшити ризик вибору неоптимальної моделі, адже різними методами комбінування було отримано кращі або не набагато гірші за найкращий окремий прогноз результати.

Загальний огляд наукових робіт щодо використання методів комбінування прогнозів за понад 50 років проведений авторами статті [18]. Підсумовуючи, вони зауважують, що попри різні запропоновані підходи до комбінування прогнозів, які теоретично мали б значно підвищити точність комбінованого прогнозу, емпіричні результати неоднозначні і часто показують, що складніші моделі досі часто

програють простому усередненню та іншим більш простим методам комбінування, що відомо як «загадка комбінування прогнозів», а однозначної відповіді на питання коли доцільніше застосовувати складніші моделі, а коли прості підходи, немає. Також звертається увага, що навіть індивідуально менш точні прогнози можуть містити корисну інформацію, тому мають враховуватись в комбінованому прогнозі. Розглядаються перспективи застосування нелінійних моделей комбінування прогнозів, заснованих на штучних нейронних мережах. В статті також згадуються програмні пакети, що знаходяться у відкритому доступі, які реалізують поширені підходи до комбінування прогнозів, такі як fable, ForecastComb і forecastHybrid для мови програмування R.

2.2 Структура інформаційної системи

Інформаційна система моделювання і прогнозування на основі методів комбінування, схема якої зображена на рис. 2.1, включає такі основні блоки: аналіз і попередня обробка часового ряду, моделювання і прогнозування, результатом якого є отримання декількох окремих прогнозних моделей, оцінка результатів моделювання і прогнозування, що виконується як для окремих моделей, так і для комбінованих прогнозів, і комбінування прогнозів різними методами, частина з яких потребує визначення вагових коефіцієнтів прогнозних моделей.

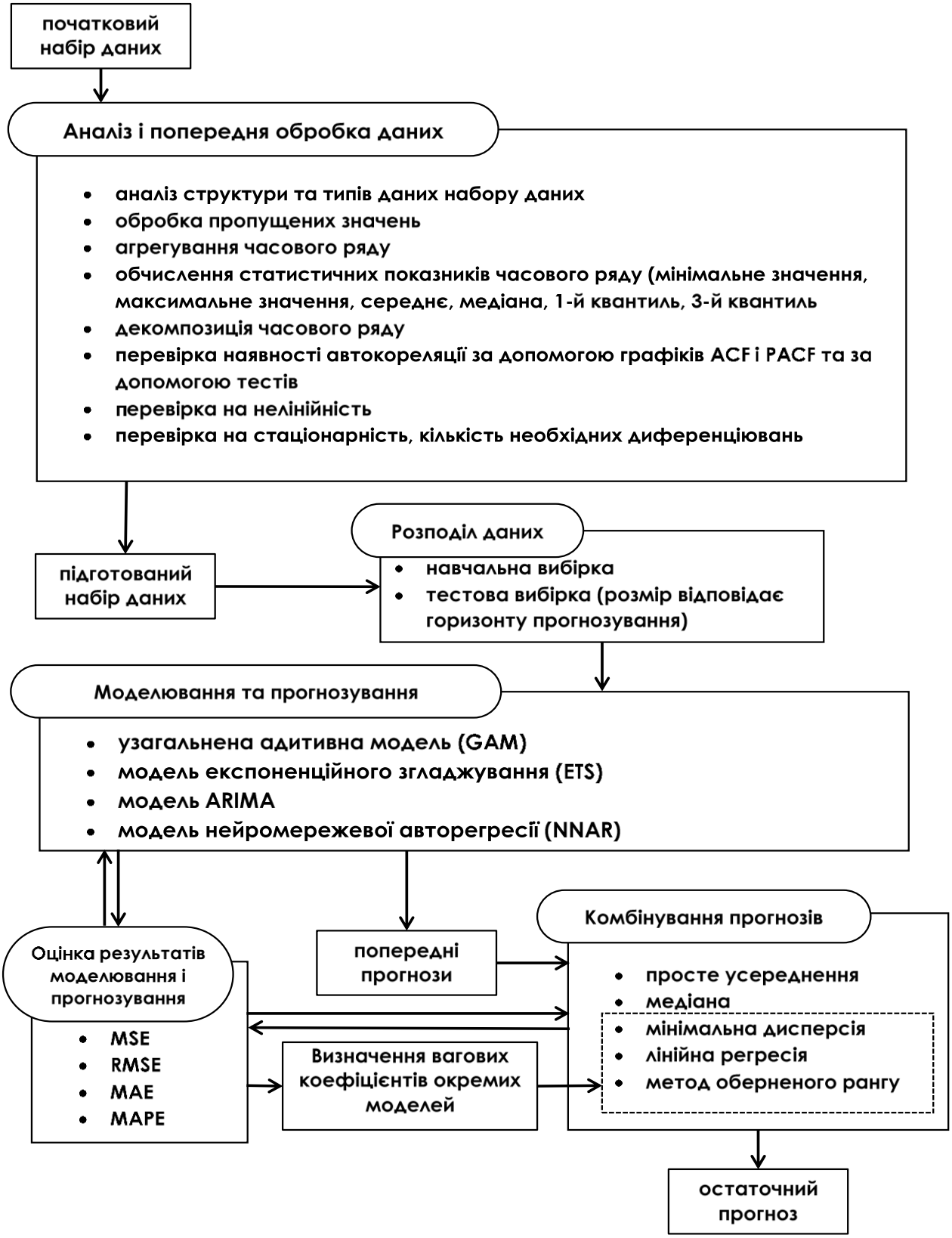

Рисунок 2.1 – Структура інформаційної системи моделювання і прогнозування на основі методів комбінування

Висновки до розділу 2

Проаналізувавши публікації на тему комбінування прогнозів було виокремлено поширені методи комбінування прогнозів і методи створення окремих прогнозних моделей, прогнози яких при цьому комбінуються.

Розглянувши публікації на тему комбінованих моделей прогнозування часових рядів можна зробити висновок, що такий підхід є ефективним у застосуванні до задачі прогнозування часових рядів, проте нема одностайної позиції щодо того, які методи комбінування прогнозів є більш ефективними. В деяких статтях складніші моделі показують кращі результати, проте деякі підтверджують більшу ефективність комбінування прогнозів шляхом простого усереднення.

Інформаційна система моделювання і прогнозування на основі методів комбінування передбачає виконання аналізу і попередньої обробки набору даних для отримання придатного для подальшої роботи часового ряду, створення окремих прогнозних моделей: узагальненої адитивної моделі, моделі ARIMA, моделі експоненційного згладжування і моделі нейромережевої авторегресії, оцінку точності отриманих моделей і комбінування прогнозів моделей кількома поширеними методами, серед яких просте усереднення, медіана, метод мінімальної дисперсії, моделі лінійної регресії та метод оберненого рангу. Комбіновані прогнози мають бути порівняні з прогнозом найкращої окремої моделі щоб зробити висновок про ефективність методів комбінування в застосуванні до обраної задачі прогнозування.

3 АНАЛІЗ ТА ПОПЕРЕДНЯ ОБРОБКА ЧАСОВОГО РЯДУ

3.1 Аналіз структури та типів даних часового ряду

Набір даних про попит на електроенергію в Україні в 2019-2024 роках взятий з сайту державної компанії оператора ринку електроенергії [38,39]. Він містить погодинні дані про обсяги купівлі і продажу електроенергії і попит на неї в МВт*год в електромережі України та ціну електроенергії починаючи з 1 липня 2019 року. Дані доступні до завантаження у вигляді файлу у форматі csv.
Завантажимо дані у змінну за допомогою функції read.csv().

```
data <- read.csv("D:/electricity_demand.csv")
```

Переглянемо структуру набору даних за допомогою функції str() (рис. 3.1).

```
> str(data)
'data.frame':   68809 obs. of  10 variables:
 $ date                     : chr  "2019-07-01" "2019-07-01" "2019-07-01" "2019-07-01" ...
 $ hour                     : int  1 2 3 4 5 6 7 8 9 10 ...
 $ energy_system            : chr  "Бурштинський п-в" "Бурштинський п-в" "Бурштинський п-в"
 $ price                    : num  939 939 939 939 939 939 939 939 2040 2040 ...
 $ amount_sales             : num  313 241 259 153 153 ...
 $ amount_purchase          : num  313 241 259 153 153 ...
 $ amount_sales_nominated   : num  313 241 259 153 153 ...
 $ amount_purchase_nominated: num  364 334 320 320 317 ...
 $ demand                   : num  -50.7 -93.3 -60.8 -167.9 -164.5 ...
 $ price_cap                : num  959 959 959 959 959 ...
```

Рисунок 3.1 – Структура набору даних

В даному випадку набір даних містить 68809 спостережень, кожне з яких характеризується 10 змінними. Серед них є дата в вигляді рядку і година представлена цілим числом, що мають бути приведені до формату даних часу (рис. 3.2).

```
data <- data %>% mutate(ymd = as.Date(date))
data <- data %>% mutate(datetime = make_datetime(year(ymd),month(ymd),d
ay(ymd),hour,0))
```

```
> str(data$datetime)
 POSIXct[1:68809], format: "2019-07-01 01:00:00" "20
19-07-01 02:00:00" "2019-07-01 03:00:00" "2019-07-01
04:00:00" ...
```

Рисунок 3.2 – Змінна datetime

Змінна demand містить попит на електроенергію, змінна energy_system позначає енергетичну підсистему і набуває лише трьох різних значень (рис. 3.3).

```
> summary(as.factor(data$energy_system))
                        Бурштинський п-в
                                   23257
                              ОЕС України
                                   23257
ОЕС України (синхронізована з ENTSO-E)
                                   22295
```

Рисунок 3.3 – Значення змінної energy_system

Ця змінна пов’язана з тим, що до 24 лютого 2022 року через технічні особливості енергосистеми України в ній існували дві відокремлені підсистеми: «Об’єднана енергетична система України» та так званий «Бурштинський енергоострів», що територіально охоплював Закарпатську та частково сусідні з нею Львівську та Івано-Франківську області, і був необхідний для експорту електроенергії в країни Європейського Союзу. 24 лютого енергосистему України було синхронізовано з європейською енергосистемою ENTSO-E [40].

Таким чином до 24 лютого 2024 набір даних містить дані про попит окремо для двох підсистем, для розрахунку загальноукраїнських показників слід виконати агрегацію значень – сумувати значення по обох зонах [38]. Структуру агрегованого набору даних наведено на рис. 3.4.

```
energy_demand <- data.frame(datetime <- unique(data$datetime), demand=0)
for(i in 1:length(energy_demand$datetime)){
  energy_demand$demand[i] = sum(data$demand[which(data$datetime==energy_dem
and$datetime[i])])
}
```

```
> str(energy_demand)
'data.frame':   46051 obs. of  2 variables:
 $ datetime....unique.data.datetime.: POSIXct, format: "2019-07-01 01:00:00" ...
 $ demand                           : num  -1348 -806 -751 -616 -521 ...
```

Рисунок 3.4 – Структура об’єкта energy_demand

3.2 Обробка пропущених значень

Для зручності подальшої роботи перетворимо набір даних на тип tsibble.

```
energy_demand <- energy_demand %>% as_tsibble(index=datetime)
```

Перевіримо наявність пропущених значень в наборі даних за допомогою функції scan_gaps() (рис. 3.5)

```
> scan_gaps(energy_demand)
# A tsibble: 5 x 1 [1h] <UTC>
  datetime
  <dttm>
1 2020-03-30 00:00:00
2 2021-03-29 00:00:00
3 2022-03-28 00:00:00
4 2023-03-27 00:00:00
5 2024-04-01 00:00:00
```

Рисунок 3.5 – Перевірка наявності пропущених значень в наборі даних

Набір даних має 5 пропущених значень (рис. 3.5). З огляду на те, що пропуски з’являються з інтервалом у рік наприкінці березня або на початку квітня можна припустити, що пов’язані вони з переведенням годинників на годину вперед під час переходу на літній час. Враховуючи, що загальний обсяг даних складає 68809 спостережень, така кількість пропусків суттєво не впливає на якість даних, проте бажано їх заповнити. Для заповнення пропущених значень в часовому ряді скористаємося стратегією LOCF, що полягає у заміні кожного пропущеного значення на останнє попереднє непропущене значення. Перевірка (рис. 3.6) показує, що пропущені значення було заповнено.

```
energy_demand <- fill_gaps(energy_demand) #заповнити значеннями NA
energy_demand <- energy_demand %>%
  mutate(demand = na_locf(demand)) #заповнити NA за методом LOCF
```

```
> scan_gaps(energy_demand)
# A tsibble: 0 x 1 [?] <UTC>
# i 1 variable: datetime <dttm>
> length(which(is.na(energy_demand)))
[1] 0
```

Рисунок 3.6 – Перевірка відсутності пропущених значень

3.3 Попередній аналіз часового ряду

Побудуємо графік часового ряду (рис. 3.7).

```
plot(x=energy_demand$datetime,y=energy_demand$demand, type="l",xlab="час",
ylab="попит (МВт*год)", main="Попит на електроенергію в енергосистемі Украї
ни",col="steelblue")
```

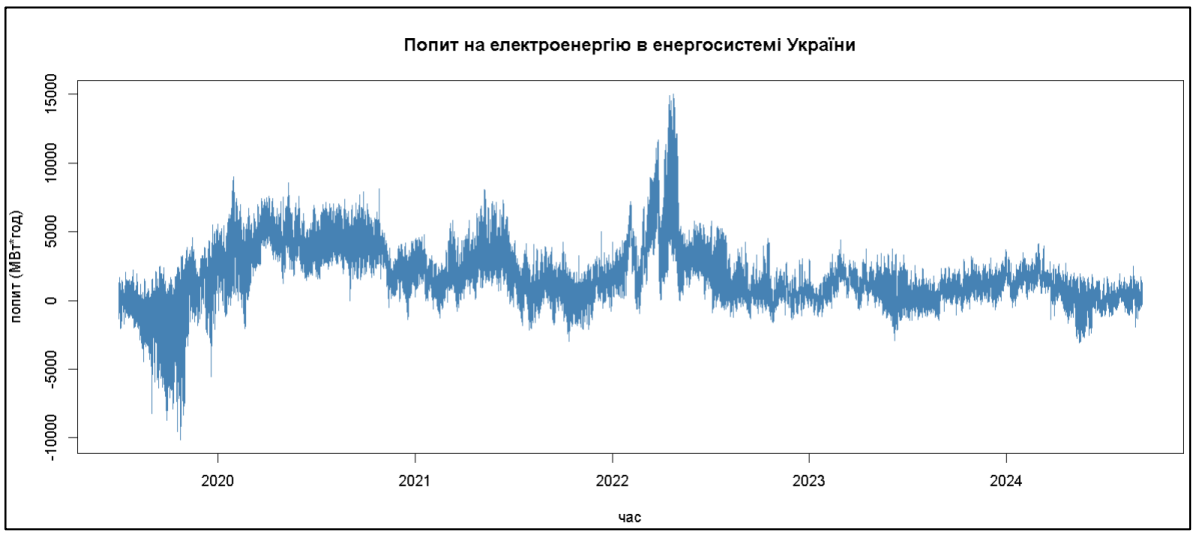

Рисунок 3.7 – Графік часового ряду

В деякі моменти часу попит може набувати від'ємних значень – це відповідає ситуації, коли обсяги продажу електроенергії перевищують обсяги купівлі.

Також можна помітити різке зростання попиту на початку 2022 року – це вочевидь пов’язано з початком повномасштабного вторгнення росії в Україну в лютому 2022 року. Наявність такої різкої і нетривалої зміни поведінки часового ряду може негативно вплинути на якість прогнозу, тому для подальшого аналізу візьмемо частину часового ряду – починаючи з 1 червня 2022 року. На рис. 3.8 наведено графік фрагменту часового ряду починаючи з червня 2022 року.

```
energy_demand <- energy_demand[which(energy_demand$datetime > make_datetime
(2022,6,1,0,0)),]
```

Рисунок 3.8 – Графік фрагмента часового ряду

Обчислимо основні статистичні показники часового ряду: мінімальне значення, 1-й квантиль (відокремлює найменші 25% значень), медіану (відокремлює менші 50% від більших 50% значень), середнє значення, 3-й квантиль (відокремлює найбільші 25% значень), максимальне значення (рис. 3.9).

```
> summary(energy_demand$demand)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-3078.1   234.8   808.3   941.8  1554.6  5795.5
```

Рисунок 3.9 – Основні статистичні показники часового ряду

Здійснимо декомпозицію часового ряду на трендову, сезонну і залишкову компоненти використовуючи метод STL() (рис. 3.10).

```
energy_demand %>%
  model(STL(demand ~ trend()+season(window=13),robust=TRUE)) %>%
  components() %>%
  autoplot()
```

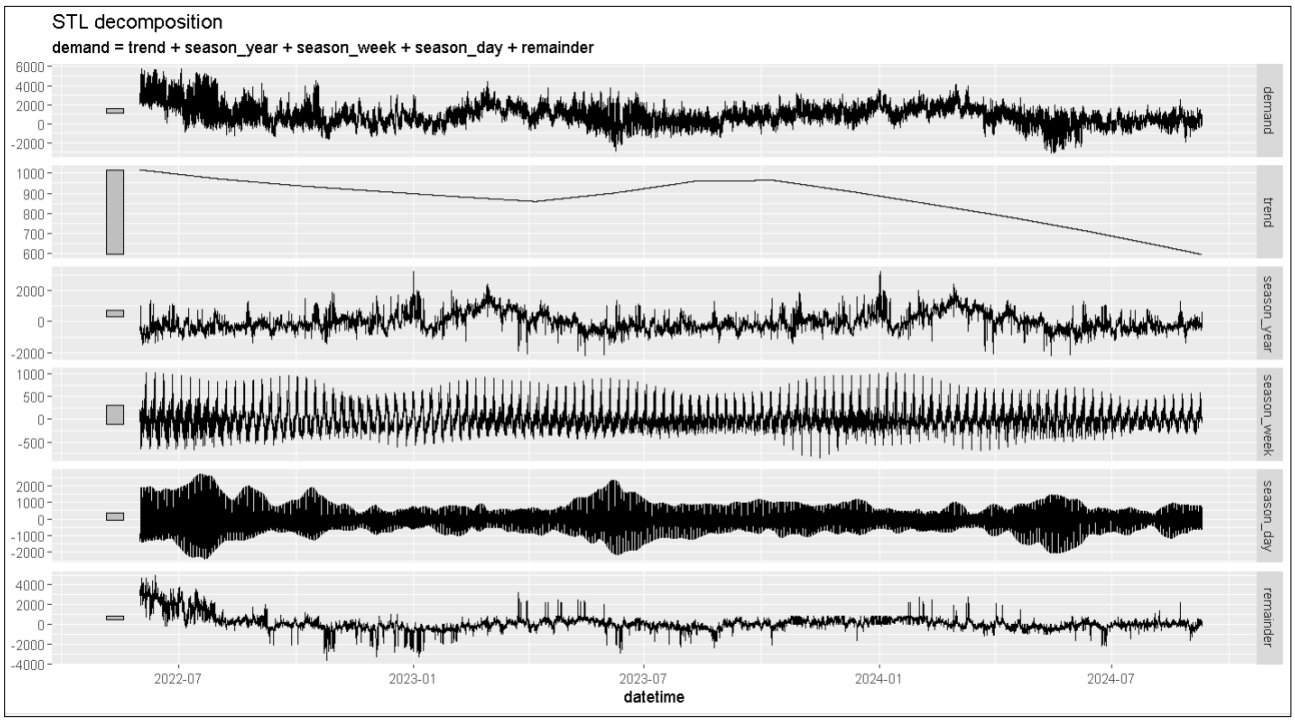

Рисунок 3.10 – Компоненти часового ряду: трендова, сезонні і залишкова

Як бачимо на рис. 3.10, було виділено три сезонні компоненти: річну, тижневу і добову, вплив річної і добової компоненти доволі значний, тижневої дещо менший. Є тренд на зменшення попиту, проте незначний. Таким чином

можна зробити висновок, що часовий ряд має складну сезонність з різними сезонними періодами.

Побудуємо графіки автокореляційної функції (ACF) та часткової автокореляційної функції (PACF) для перевірки наявності у часового ряду автокореляції.

```
acf(energy_demand$demand)
pacf(energy_demand$demand)
```

Як бачимо на рис. 3.11, кореляція дуже суттєва, причому вона періодично зростає і спадає з періодом 24, що пояснюється наявністю вираженої добової сезонності. Графіка PACF (рис. 3.12) також демонструє помітну автокореляцію і наявність добової сезонності.

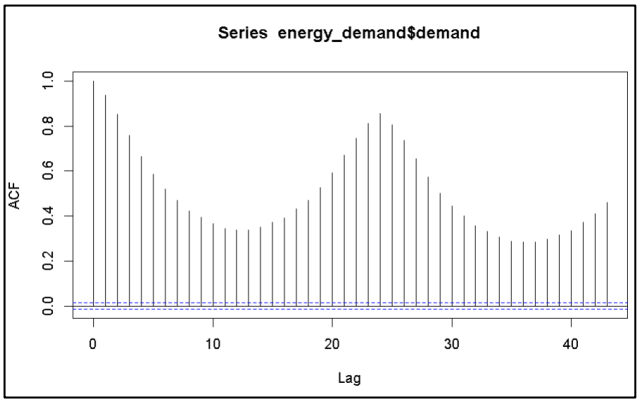

Рисунок 3.11 – Графік автокореляційної функції для часового ряду (корелограма)

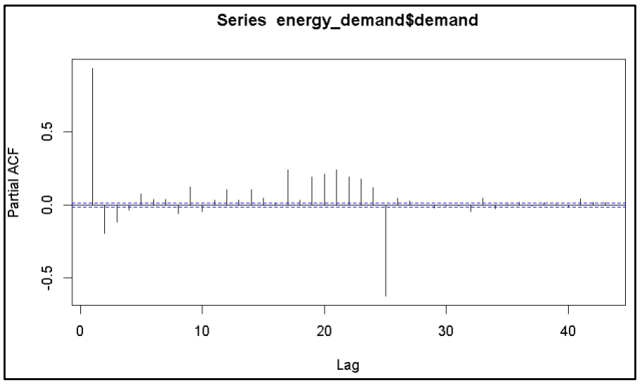

Рисунок 3.12 – Графік часткової автокореляційної функції

Також можна провести тести на автокореляцію Дарбіна-Вотсона та Бройша-Ґотфрі. Ці тести вимагають спочатку створити модель лінійної регресії. Результати тестів наведено на рис. 3.13.

```
model <- lm(energy_demand$demand ~ energy_demand$datetime)#модель лінійної
регресії
dwtest(model)#тест Дарбіна-Вотсона
bgtest(model)#тест Бройша-Ґотфрі
```

```
> dwtest(model)#тест Дарбіна-Вотсона

        Durbin-Watson test

data:  model
DW = 0.13364, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0

> bgtest(model)#тест Бройша-Ґотфрі

        Breusch-Godfrey test for serial correlation of order up to 1

data:  model
LM test = 17388, df = 1, p-value < 2.2e-16
```

Рисунок 3.13 – Результати тестів на автокореляцію

В обох випадках значення p набагато менші за 0,05 (рис. 3.13), отже є підстави вважати, що в часовому ряді наявна автокореляція.

Виконані тести на нелінійність (набір таких тестів виконує функція nonlinearityTest()) вказують на нелінійність часового ряду, оскільки значення p менші за 0,05 , а тест Маклеода-Лі, оскільки значення p менше за 0,05 вказує на гетероскедастичність – тобто розкид залишків змінюється по мірі зміни рівня часового ряду (рис. 3.14).

```
> nonlinearityTest(energy_demand$demand)
                ** Teraesvirta's neural network test  **
                Null hypothesis: Linearity in "mean"
                X-squared =  173.6739  df =  2  p-value =  0

                ** White neural network test  **
                Null hypothesis: Linearity in "mean"
                X-squared =  141.4728  df =  2  p-value =  0

                ** Keenan's one-degree test for nonlinearity  **
                Null hypothesis: The time series follows some AR process
                F-stat =  41.27541  p-value =  1.3522e-10

                ** McLeod-Li test  **
                Null hypothesis: The time series follows some ARIMA process
                Maximum p-value =  0

                ** Tsay's Test for nonlinearity **
                Null hypothesis: The time series follows some AR process
                F-stat =  4.684975  p-value =  0

                ** Likelihood ratio test for threshold nonlinearity **
                Null hypothesis: The time series follows some AR process
                Alternativce hypothesis: The time series follows some TAR process
                X-squared =  387.6993  p-value =  0
```

Рисунок 3.14 – Результати тестів на нелінійність часового ряду

Виконаємо тести на кількість необхідних для отримання стаціонарного часового ряду операцій диференціювання і сезонного диференціювання (рис. 3.15).

```
> unitroot_ndiffs(energy_demand$demand)
ndiffs
     1
> unitroot_nsdiffs(energy_demand$demand)
nsdiffs
      0
```

Рисунок 3.15 – Тести на кількість необхідних операцій диференціювання

Отримано, що необхідно виконати диференціювання 1 раз, а сезонні диференціювання не потрібні. Виконаємо диференціювання часового ряду і побудуємо графік диференційованого часового ряду (рис. 3.16).

```
energy_demand_diff <- diff(energy_demand$demand)
```

```
plot(x=energy_demand$datetime[2:length(energy_demand$datetime)],
     y=energy_demand_diff, type="l",
     xlab="час", ylab="diff(demand)",
     main="Диференційований часовий ряд",col="steelblue")
```

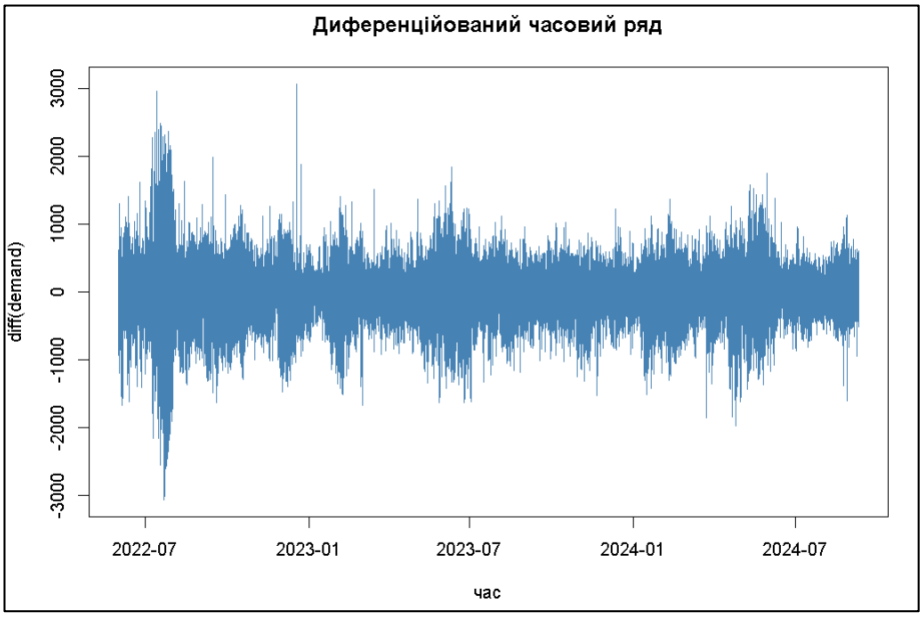

Рисунок 3.16 – Графік диференційнованого часового ряду

Оскільки прогноз щодо попиту на електроенергію доцільно здійснювати на кілька днів, а не в межах доби, а також враховуючи значний розмір наявної вибірки, навіть за умови взяття лише частини даних, то має сенс агрегувати дані про попит за датами, отримавши таким чином часовий ряд, що міститиме серденій попит на годину для кожної дати.

```
energy_demand_daily <- energy_demand %>% group_by_key() %>% index_by(date =
~ as_date(.)) %>% summarize(avg_demand = mean(demand))

plot(x=energy_demand_daily$date,y=energy_demand_daily$avg_demand, type="l",
xlab="дата", ylab="середній за годину попит (МВт*год)", main="Попит на елек
троенергію в енергосистемі України",col="steelblue")
```

На рис. 3.17 наведено графік агрегованого часового ряду

Рисунок 3.17 – Графік агрегованого часового ряду

Основні статистичні показники агрегованого часового ряду наведено на рис. 3.18.

```
summary(energy_demand_daily$avg_demand)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-790.5   365.2   824.7   941.2  1418.8  4134.5
```

Рисунок 3.18 – Статистичні показники агрегованого часового ряду

Виконаємо декомпозицію агрегованого часового ряду на компоненти (рис. 3.19).

```
energy_demand_daily %>%
  model(STL(avg_demand~trend()+season(window=13),robust=TRUE)) %>%
  components() %>% autoplot()
```

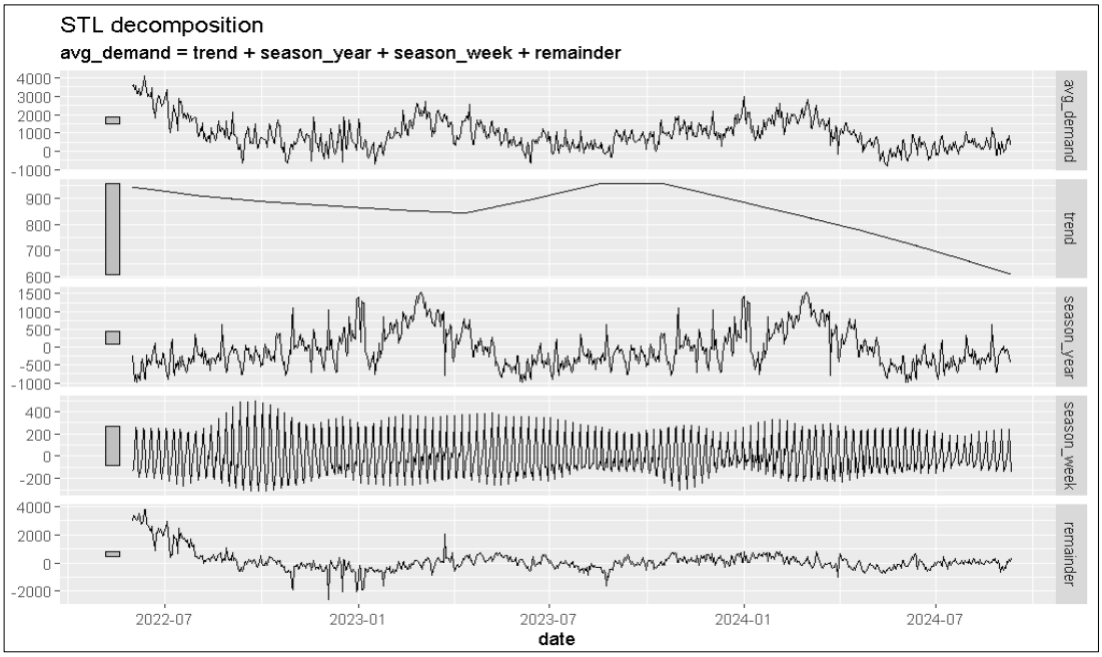

Рисунок 3.19 – Результати декомпозиції агрегованого часового ряду

Агрегований часовий ряд має сильну річну сезонність і дещо меншу тижневу, вплив тренду слабкий.

Побудуємо графіки автокореляційної та часткової автокореляційної функцій. На графіках (рис. 3.20, 3.21) можна помітити високу автокореляцію, і наявність тижневої сезонності

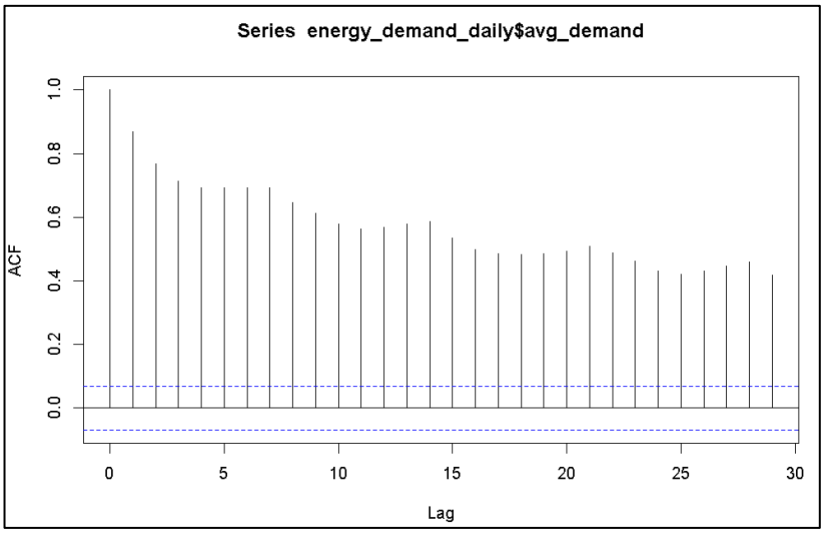

Рисунок 3.20 – Графік автокореляційної функції

Графік часткової автокореляційної функції (рис. 3.21) підтверджує наявність помітної автокореляції зі спостереженнями наступного тижня.

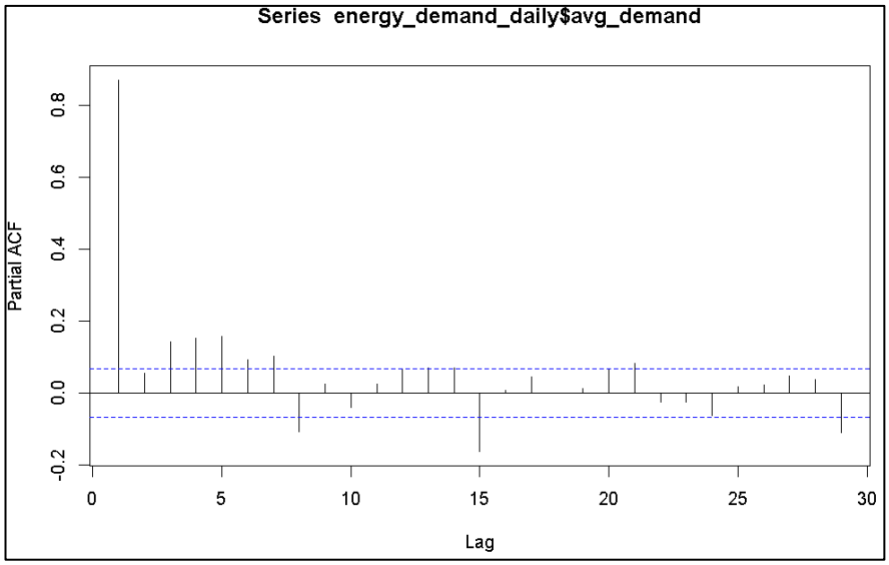

Рисунок 3.21 – Графік часткової автокореляційної функції

Тести на автокореляцію підтверджують її наявність (рис. 3.22).

```
> dwtest(model)#тест Дарбіна-Вотсона

        Durbin-Watson test

data:  model
DW = 0.27634, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0

> bgtest(model)#тест Бройша-Ґотфрі

        Breusch-Godfrey test for serial correlation of order up to
        1

data:  model
LM test = 612.2, df = 1, p-value < 2.2e-16
```

Рисунок 3.22 – Результати тестів на наявність автокореляції

Тести на нелінійність (рис. 3.23) вказують на нелінійність та гетероскедастичність часового ряду.

```
> nonlinearityTest(energy_demand_daily$avg_demand)
                ** Teraesvirta's neural network test  **
                Null hypothesis: Linearity in "mean"
                X-squared =  12.15562  df =  2  p-value =  0.002293193

                ** White neural network test  **
                Null hypothesis: Linearity in "mean"
                X-squared =  9.11184  df =  2  p-value =  0.01050483

                ** Keenan's one-degree test for nonlinearity  **
                Null hypothesis: The time series follows some AR process
                F-stat =  0.9188089  p-value =  0.3380876

                ** McLeod-Li test  **
                Null hypothesis: The time series follows some ARIMA process
                Maximum p-value =  0

                ** Tsay's Test for nonlinearity **
                Null hypothesis: The time series follows some AR process
                F-stat =  1.256644  p-value =  0.01351194

                ** Likelihood ratio test for threshold nonlinearity **
                Null hypothesis: The time series follows some AR process
                Alternativce hypothesis: The time series follows some TAR process
                X-squared =  42.4911  p-value =  0.3058105
```

Рисунок 3.23 – Результати тестів на нелінійність

Щоб зробити часовий ряд стаціонарним достатньо одноразового диференціювання, сезонне диференціювання не потрібне (рис. 3.24). На рис. 3.25 наведено графік диференційованого часового ряду.

```
> unitroot_ndiffs(energy_demand_daily$avg_demand)
ndiffs
     1
> unitroot_nsdiffs(energy_demand_daily$avg_demand)
nsdiffs
      0
```

Рисунок 3.24 – Результати тестів на кількість диференціювань

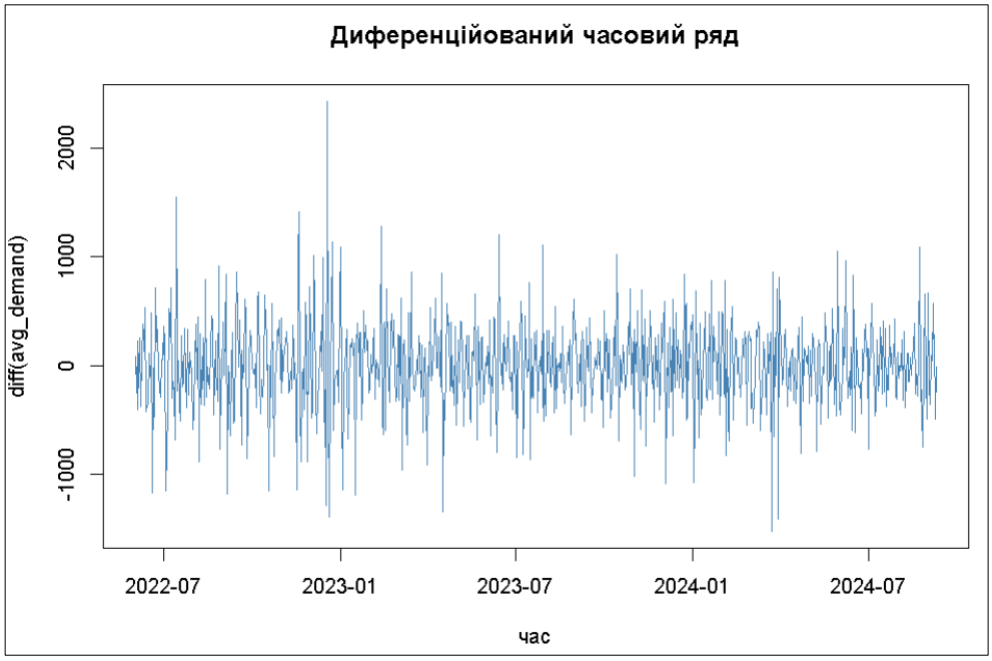

Рисунок 3.25 – Графік диференційованого агрегованого часового ряду

Для подальшої роботи розіб’ємо часовий ряд на дві вибірки: навчальну і тестову. Тестова вибірка міститиме спостереження за останні 30 днів, всі попередні спостереження міститимуться в навчальній вибірці, таким чином прогноз виконуватиметься на місяць вперед на основі моделі, побудованої за навчальною вибіркою, після чого прогнозні значення порівнюватимуться з реальними. На рис. 3.36 наведено розміри навчальної і тестової вибірки.

```
train_set <- head(energy_demand_daily,length(energy_demand_daily$avg_demand
)-30)
test_set <- tail(energy_demand_daily,30)
length(train_set$avg_demand)
length(test_set$avg_demand)
```

```
> length(train_set$avg_demand)
[1] 803
> length(test_set$avg_demand)
[1] 30
```

Рисунок 3.26 – Кількість спостережень навчальної і тестової вибірки

Висновки до розділу 3

В цьому розділі було проаналізовано часовий ряд, що містить дані про попит на електроенергію в енергомережі України за останні роки. Було обрано фрагмент даних для аналізу, розглянуто його статистичні характеристики. Було виконано декомпозицію часового ряду і виявлено помітну сезонність з різними періодами: річну, тижневу, добову. Було виконано агрегацію часового ряду по датам з метою зменшення обсягу аналізованих даних, а також з точки зору доцільності виконання прогнозу на деяку кількість днів вперед, а не погодинно. Було виявлено помітну автокореляцію і визначено кількість необхідних диференціювань.

Повний лістинг коду аналізу і попередньої обробки даних наведено в додатку А.

4 СТВОРЕННЯ ОКРЕМИХ ПРОГНОЗНИХ МОДЕЛЕЙ І КОМБІНУВАННЯ ПРОГНОЗІВ

4.1 Узагальнена адитивна модель

Використовуючи бібліотеку prophet побудуємо кілька моделей: з параметрами за замовчуванням, з більшою кількістю вузлових точок тренду, з меншою кількістю вузлових точок тренду, з більшим впливом сезонних компонент і з мультиплікативною сезонністю.

```
library(prophet)#підключення бібліотеки prophet
train_set_prophet <- data.frame(ds=train_set$date,y=train_set$avg_demand)
test_set_prophet <- data.frame(ds=test_set$date,y=test_set$avg_demand)
prophet_m1 <- prophet(train_set_prophet)
forecast_prophet_m1 <- predict(prophet_m1, make_future_dataframe(prophet_m1
,periods = 30))
prophet_m2 <- prophet(train_set_prophet, changepoint.prior.scale = 0.2)
forecast_prophet_m2 <- predict(prophet_m2, make_future_dataframe(prophet_m2
,periods = 30))
prophet_m3 <- prophet(train_set_prophet, changepoint.prior.scale = 0.02)
forecast_prophet_m3 <- predict(prophet_m3, make_future_dataframe(prophet_m3
,periods = 30))
prophet_m4 <- prophet(train_set_prophet, seasonality.prior.scale = 20)
forecast_prophet_m4 <- predict(prophet_m4, make_future_dataframe(prophet_m4
,periods = 30))
prophet_m5 <- prophet(train_set_prophet, seasonality.mode = "multiplicative
")
forecast_prophet_m5 <- predict(prophet_m5, make_future_dataframe(prophet_m5
,periods = 30))
```

Для оцінки побудованих моделей можна скористатися методом перехресної перевірки, реалізованим в бібліотеці prophet. В цьому методі будується прогноз на заданий прогнозний горизонт за частиною навчальних даних, після чого отриманий

прогноз порівнюється зі справжніми даними. Далі аналогічно будується прогноз за більшою підмножиною навчальних даних і порівнюється з відповідними справжніми даними, і так доки не досягнуто кінця вибірки. Таким чином метод перехресної перевірки багаторазово імітує перевірку прогнозної моделі на тестовій вибірці, після чого отримані результати підсумовуються [41].

```
prophet_m1_performance <- performance_metrics(
  cross_validation(prophet_m1,horizon=30,unit="days"),
  metrics = c("mse","rmse","mae","coverage"), rolling_window = 1)
```

На рис. 4.1 наведено показники ефективності моделі, розраховані методом перехресної перевірки: MSE, RMSE, MAE і покриття (частка значень, що знаходяться в довірчих межах прогнозу). Розрахунок MAPE проводити недоцільно, оскільки багато значень вибірки близькі до нуля. Очевидно що друга модель виглядає найбільш оптимальною.

```
> cross_validation_performance
  Model       MSE      RMSE      MAE  coverage
1    m1  701170.7  837.3593 698.4296 0.4154762
2    m2  492283.6  701.6293 561.8430 0.5511905
3    m3 1096471.6 1047.1254 878.8227 0.3595238
4    m4  704413.9  839.2937 699.1083 0.4166667
5    m5 1749510.0 1322.6904 828.4709 0.5011905
```

Рисунок 4.1 – Результати перехресної перевірки

Щоб остаточно визначити оптимальну модель здійснимо перевірку на тестовій вибірці.

```
performanceMetrics <- function(predicted,actual) {
  return (data.frame(
    mse=mean((predicted-actual)^2),rmse=sqrt(mean((predicted-actual)^2)),
    mae=mean(abs(predicted-actual))))}
predictions_prophet_m1 <- tail(forecast_prophet_m1$yhat,30)
prophet_m1_performance=performanceMetrics(predictions_prophet_m1,test_set$a
vg_demand)
```

Перевірка на тестовій вибірці, результати якої наведено на рис. 4.2 також показує оптимальність другої моделі.

```
> prophet_test_set_performance
  Model      MSE     RMSE      MAE
1    m1 628387.8 792.7092 724.5165
2    m2 157228.5 396.5205 324.3262
3    m3 500017.0 707.1188 638.5019
4    m4 592512.7 769.7485 700.4694
5    m5 308917.3 555.8033 447.9766
```

Рисунок 4.2 – Результат перевірки точності моделей на тестовій вибірці

На рис. 4.3 наведено графік побудованої моделі і прогнозу на її основі, чорні крапки позначають елементи навчальної вибірки, червоні – тестової. На рис. 4.4 детальніше відображено результат прогнозування: відтінками синього позначено прогнозні значення і 80%-вий довірчий інтервал, червоним відображено реальні значення з тестової вибірки.

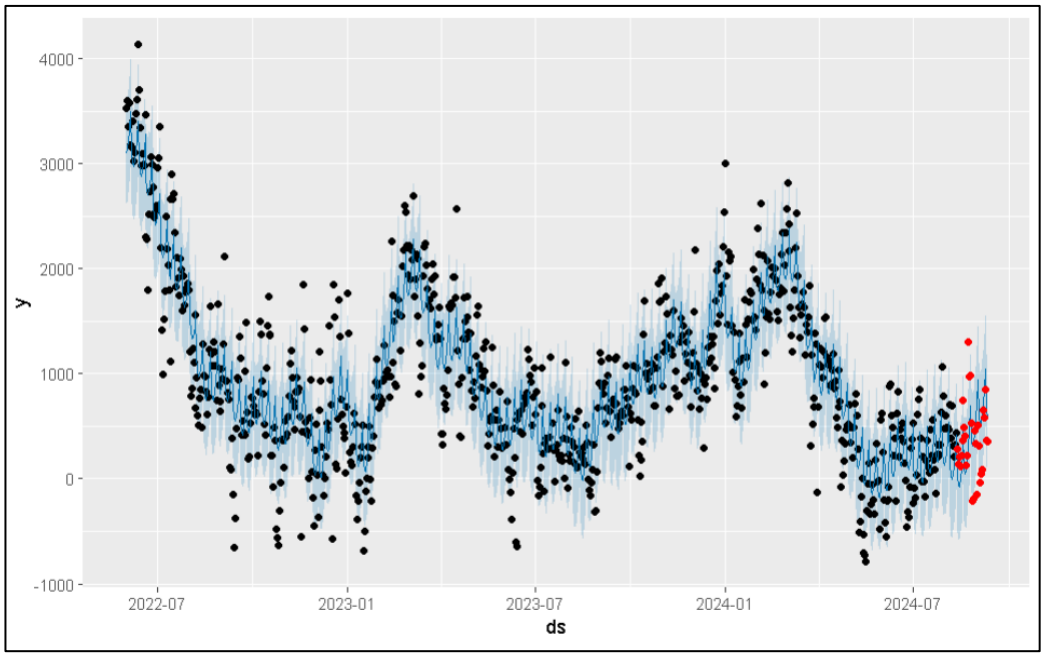

Рисунок 4.3 – Графік створеної моделі і побудованого прогнозу

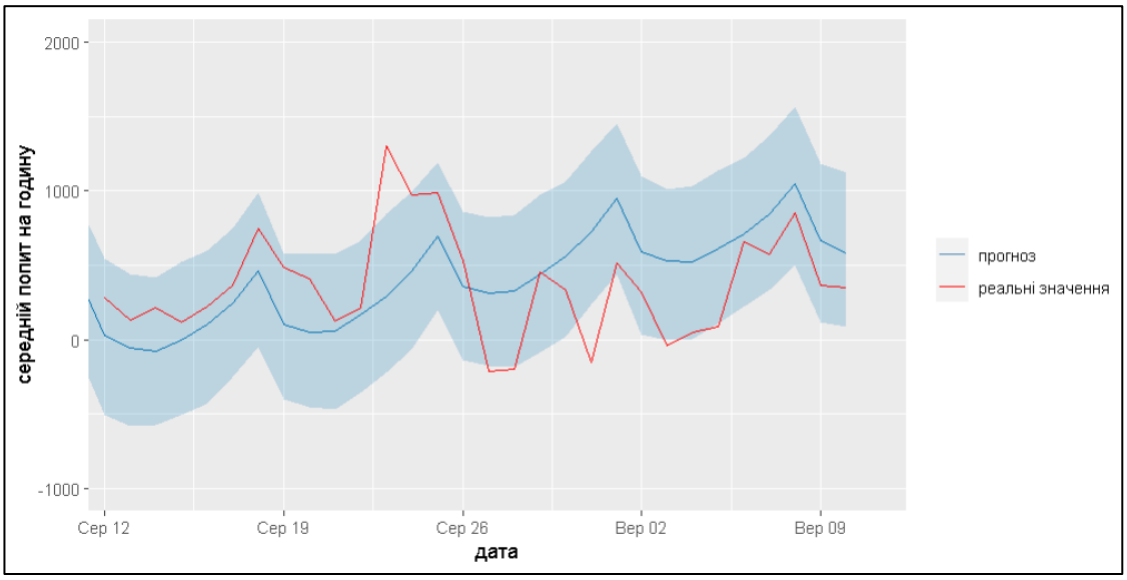

Рисунок 4.4 – Графік прогнозованого моделлю prophet і реального попиту

4.2 Модель експоненційного згладжування

Для створення прогнозних моделей на основі методу експоненційного згладжування використовувалися функції бібліотеки forecast. Було побудовано 4 моделі: просте експоненційне згладжування, експоненційне згладжування з трендом – метод Хольта, експоненційне згладжування з трендом і сезонністю – метод Хольта-Вінтерса, і експоненційне згладжування з адитивними трендом, сезонністю і помилками.

```
train_set_ts <- as.ts(train_set)#перетворення навчальної вибірки на тип ts
#моделі
ses_model <- ses(train_set_ts, h=30) #просте експоненційне згладжування
holt_model <- holt(train_set_ts, h=30) #метод Хольта
hw_model <- hw(train_set_ts, h=30) #метод Хольта-Вінтерса
aaa_model <- predict(ets(train_set_ts,"AAA"), h=30) #метод Хольта-Вінтерса
з адитивними помилками
```

Перш ніж оцінювати ефективність моделей на тестовій вибірці можна переглянути оцінки, розраховані при підгонці моделей. Оцінка точності моделей на навчальній вибірці показує, що дві останні моделі є більш точними (рис. 4.5).

```
#оцінка точності моделей на навчальній вибірці
models_accuracy <- rbind(accuracy(ses_model),accuracy(holt_model),accuracy(
hw_model),accuracy(aaa_model))
rownames(models_accuracy) <- c("ses","holt","hw","aaa")
```

```
> models_accuracy
                ME     RMSE      MAE
ses   -5.2213463 406.8511 309.4717
holt   0.1440086 406.8469 309.4385
hw   -15.3710413 366.2524 268.1752
aaa  -15.0038479 366.6633 268.3388
```

Рисунок 4.5 – Оцінка точності моделей експоненційного згладжування на навчальній вибірці

Переглянемо значення інформаційних критеріїв AIC, AICc і BIC, підраховані при підгонці моделей. Менші значення для останніх двох моделей свідчать, що вони є більш оптимальними (рис. 4.6).

```
#інформаційні критерії AIC, AICc і BIC
information_criteria <- data.frame(
  Model=c("ses","holt","hw","aaa"),
  AIC=c(ses_model$model$aic, holt_model$model$aic, hw_model$model$aic,aaa_m
odel$model$aic),
  AICc=c(ses_model$model$aicc, holt_model$model$aicc, hw_model$model$aicc,a
aa_model$model$aicc),
  BIC=c(ses_model$model$bic, holt_model$model$bic, hw_model$model$bic,aaa_m
odel$model$bic)
)
```

```
> information_criteria
  Model      AIC     AICc      BIC
1   ses 15026.32 15026.35 15040.38
2  holt 15030.30 15030.37 15053.74
3    hw 14875.49 14875.88 14931.75
4   aaa 14877.29 14877.68 14933.55
```

Рисунок 4.6 – Значення інформаційних критеріїв для моделей

Щоб остаточно вибрати оптимальну модель, здійснимо перевірку на тестовій вибірці – результати наведено на рис. 4.7.

```
ets_models_performance <- rbind(
  performanceMetrics(ses_model$mean,test_set$avg_demand),
  performanceMetrics(holt_model$mean,test_set$avg_demand),
  performanceMetrics(hw_model$mean,test_set$avg_demand),
  performanceMetrics(aaa_model$mean,test_set$avg_demand))
rownames(ets_models_performance) <- c("ses","holt","hw","aaa")
```

```
> ets_models_performance
           mse     rmse      mae
ses  125493.7 354.2508 277.4516
holt 127238.3 356.7047 279.0931
hw   135878.5 368.6171 290.1608
aaa  127475.5 357.0371 276.2925
```

Рисунок 4.7 – Оцінка точності моделей експоненційного згладжування на тестовій вибірці

Отже за даними, наведеними на рис. 4.5-4.7 можна зробити висновок про оптимальність останньої моделі – з адитивними трендом, сезонністю і помилками. На рис. 4.8 наведено графік прогнозу цієї моделі з 80%-м і 95%-м прогнозними інтервалами, а на рис. 4.9 наведено графік прогнозованих значень і реальних значень попиту.

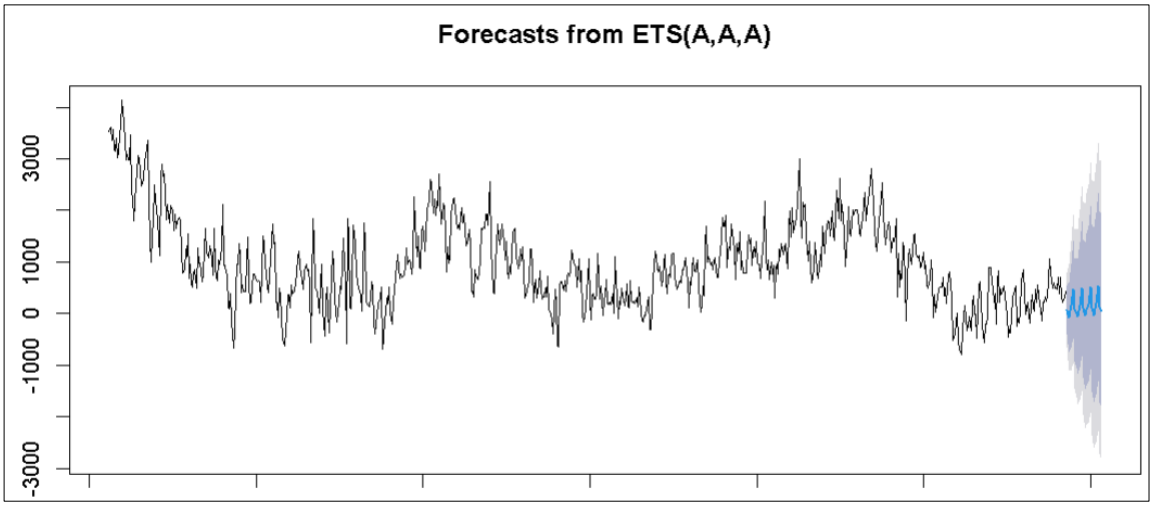

Рисунок 4.8 – Прогноз моделі експоненційного згладжування

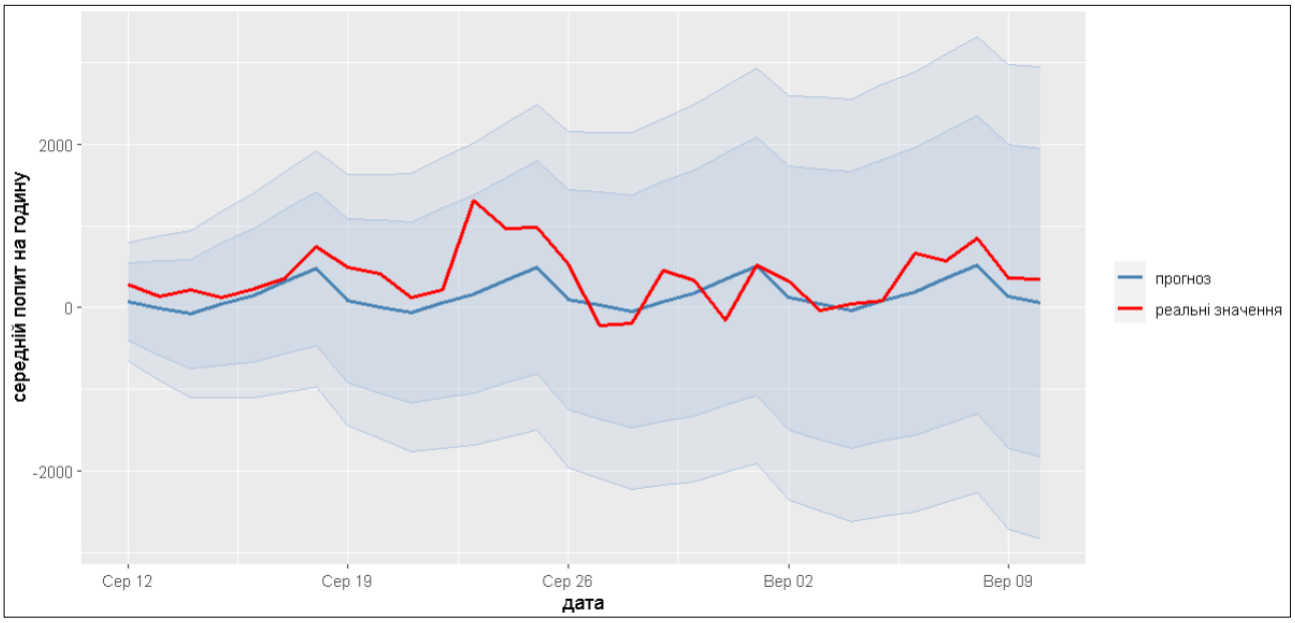

Рисунок 4.9 – Графік прогнозованого моделлю ETS(A,A,A) і реального попиту

4.3 Модель ARIMA

Для побудови моделі ARIMA скористаємось функцією auto.arima() для автоматичного підбору параметрів моделі.

```
arima_m1 <- forecast(auto.arima(train_set_ts),h=30)
```

Було побудовано модель ARIMA(1,1,2)(0,0,2)$_7$, тобто несезонна частина має порядок авторегресійної частини 1, порядок диференціювання 1 і порядок частини з ковзним середнім 2, сезонна частина відображає тижневу сезонність і має порядок частини з ковзним середнім 2. На рис. 4.10 наведено опис моделі.

```
> summary(arima_m1$model)
Series: train_set_ts
ARIMA(1,1,2)(0,0,2)[7]

Coefficients:
         ar1     ma1      ma2    sma1    sma2
      0.5173  -0.773  -0.1203  0.1412  0.1527
s.e.  0.0696   0.074   0.0545  0.0369  0.0316

sigma^2 = 139982:  log likelihood = -5887.5
AIC=11787   AICc=11787.1   BIC=11815.12

Training set error measures:
                    ME     RMSE      MAE         MPE
Training set -13.26252 372.7411 281.0717 -0.4840935
```

Рисунок 4.10 – Інформація про побудовану функцією auto.arima() модель

Побудуємо ще декілька моделей, які трохи відрізнятимуться від побудованої.

```
arima_m2 <- forecast(Arima(train_set_ts,order=c(2,1,2),season=c(0,0,2)),h=30)
arima_m3 <- forecast(Arima(train_set_ts,order=c(1,1,2),season=c(2,0,2)),h=30)
arima_m4 <- forecast(Arima(train_set_ts,order=c(2,1,2),season=c(1,0,2)),h=30)
arima_m5 <- forecast(Arima(train_set_ts,order=c(2,1,2),season=c(2,1,2)),h=30)
```

Порівняємо точність отриманих моделей на навчальній і тестовій вибірках (рис. 4.11-4.12).

```
#точність моделі на навчальній вибірці
arima_models_accuracy <- rbind(
  accuracy(arima_m1),
  accuracy(arima_m2),
  accuracy(arima_m3),
  accuracy(arima_m4),
  accuracy(arima_m5)
)
rownames(arima_models_accuracy) <- c(as.character(arima_m1$model), as.chara
cter(arima_m2$model),as.character(arima_m3$model), as.character(arima_m4$mo
del),as.character(arima_m5$model))
arima_models_accuracy

#точність моделі на тестовій вибірці
arima_models_performance <- rbind(
  performanceMetrics(arima_m1$mean,test_set$avg_demand),
  performanceMetrics(arima_m2$mean,test_set$avg_demand),
  performanceMetrics(arima_m3$mean,test_set$avg_demand),
  performanceMetrics(arima_m4$mean,test_set$avg_demand),
  performanceMetrics(arima_m5$mean,test_set$avg_demand)
)
rownames(arima_models_performance) <- c(as.character(arima_m1$model), as.ch
aracter(arima_m2$model),as.character(arima_m3$model), as.character(arima_m4
$model),as.character(arima_m5$model))
arima_models_performance
```

	ME	RMSE	MAE
ARIMA(1,1,2)(0,0,2)[7]	-13.262525	372.7411	281.0717
ARIMA(2,1,2)(0,0,2)[7]	-13.185746	372.7355	281.0588
ARIMA(1,1,2)(2,0,2)[7]	8.999448	352.5602	260.6590
ARIMA(2,1,2)(1,0,2)[7]	9.303075	353.1975	260.7882
ARIMA(2,1,2)(2,1,2)[7]	21.377046	350.7293	259.5981

Рисунок 4.11 – Показники точності моделі на навчальній вибірці

	mse	rmse	mae
ARIMA(1,1,2)(0,0,2)[7]	134552.86	366.8145	291.9004
ARIMA(2,1,2)(0,0,2)[7]	134702.54	367.0184	292.1864
ARIMA(1,1,2)(2,0,2)[7]	96497.25	310.6401	231.0065
ARIMA(2,1,2)(1,0,2)[7]	93986.56	306.5723	228.7214
ARIMA(2,1,2)(2,1,2)[7]	97004.87	311.4561	228.8635

Рисунок 4.12 – Показники точності моделі

Отже в результаті перевірки на тестовій вибірці найоптимальнішою є модель ARIMA(2,1,2)(1,0,2)$_7$. На рис. 4.13 наведено прогноз зроблений цією моделлю з 80%-м і 95%-м прогнозними інтервалами, а на рис. 4.14 цей прогноз та реальні дані.

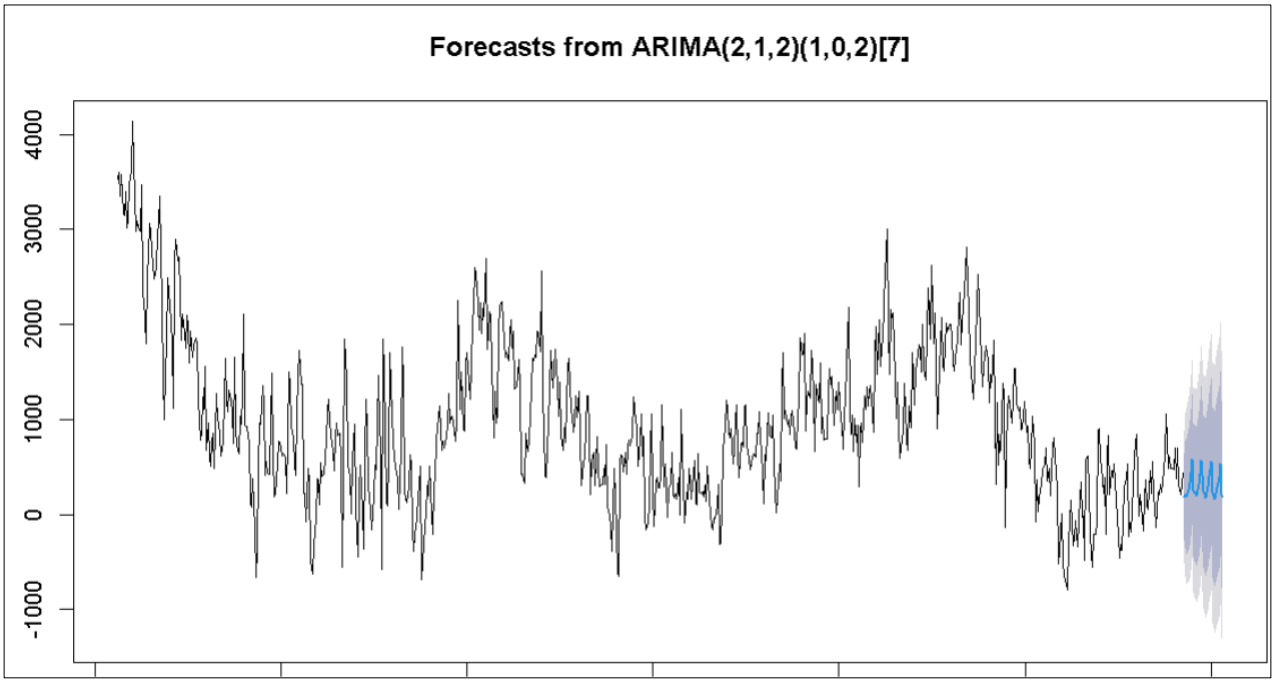

Рисунок 4.13 – Прогноз, побудований моделлю ARIMA

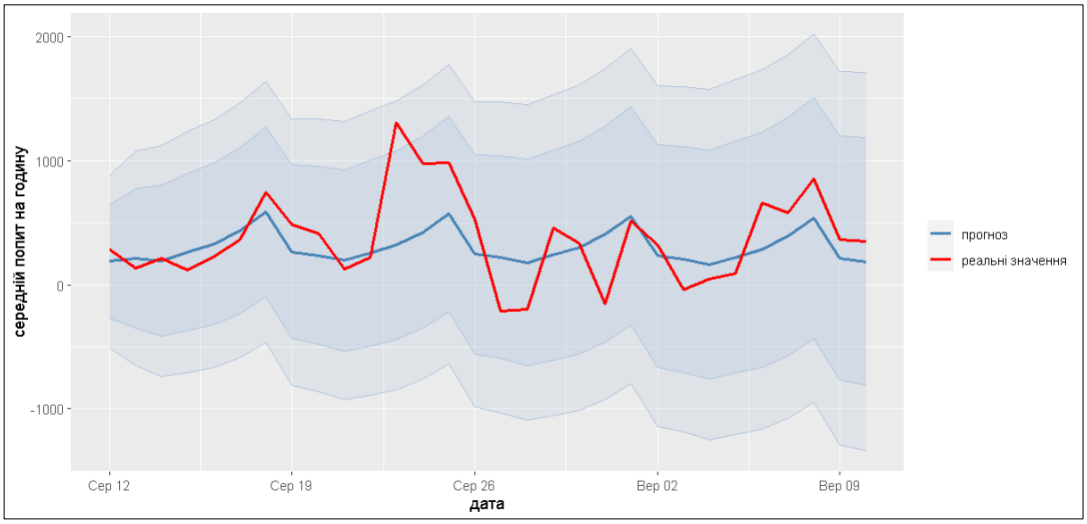

Рисунок 4.14 - Графік прогнозованого моделлю ARIMA і реального попиту

4.4 Нейромережеві моделі

Моделі на основі нейромережевої авторегресії дозволяє побудувати функція nnetar(). Параметри функції дозволяють встановити кількість входів несезонної і сезонної частини моделі та кількість моделей, які буде побудовано (результуючий прогноз буде в такому разі усередненням прогнозів декількох моделей). Збільшення значень цих параметрів призводить до значного зростання часу навчання моделі. На рис. 4.15 наведено оцінки точності побудованих моделей на тестовій вибірці.

```
nn_m1 <- nnetar(train_set_ts)
nn_m2 <- nnetar(train_set_ts, p = 25, P = 25, MaxNWts=1500)
nn_m3 <- nnetar(train_set_ts, p = 50, P = 50,repeats = 50,MaxNWts = 5000)
nn_m4 <- nnetar(train_set_ts, p = 100, P = 100, MaxNWts = 20000)
```

```
> nn_models_performance
        mse     rmse      mae
1 283759.35 532.6907 443.0852
2 417616.99 646.2329 487.7922
3 431768.49 657.0909 533.0027
4 110804.56 332.8732 256.0339
```

Рисунок 4.15 – Точність побудованих моделей на тестовій вибірці

Таким чином найбільш оптимальною є модель з найбільшою кількістю входів і найбільшою кількістю побудованих нейромережевих моделей. Побудований моделлю прогноз наведено на рис. 4.16, 4.17.

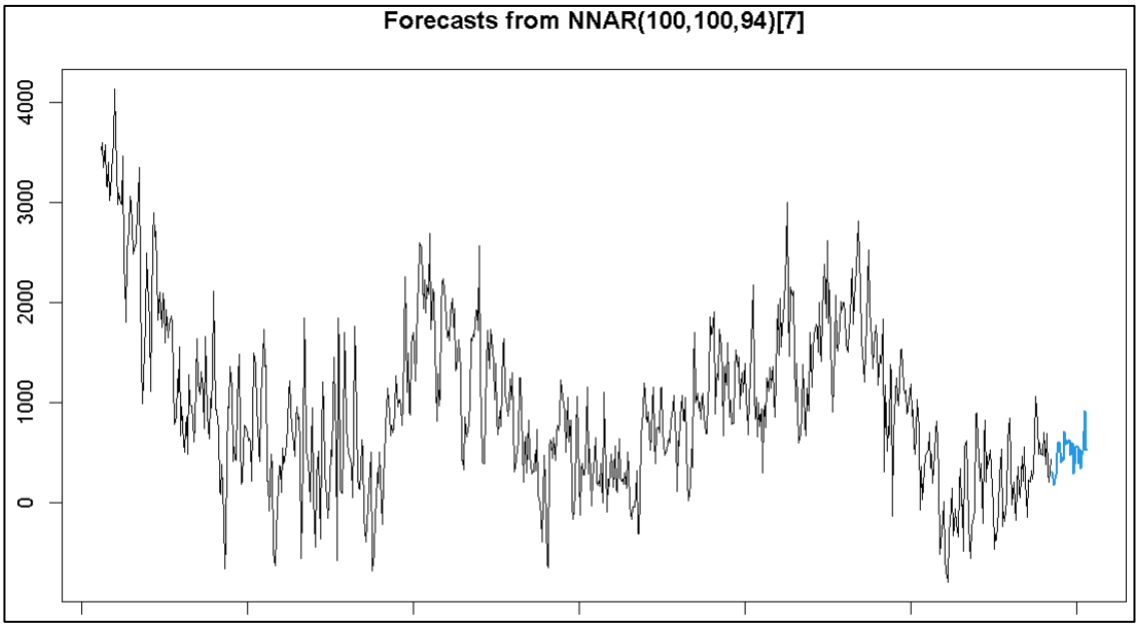

Рисунок 4.16 – Прогноз, побудований нейромережевою моделлю

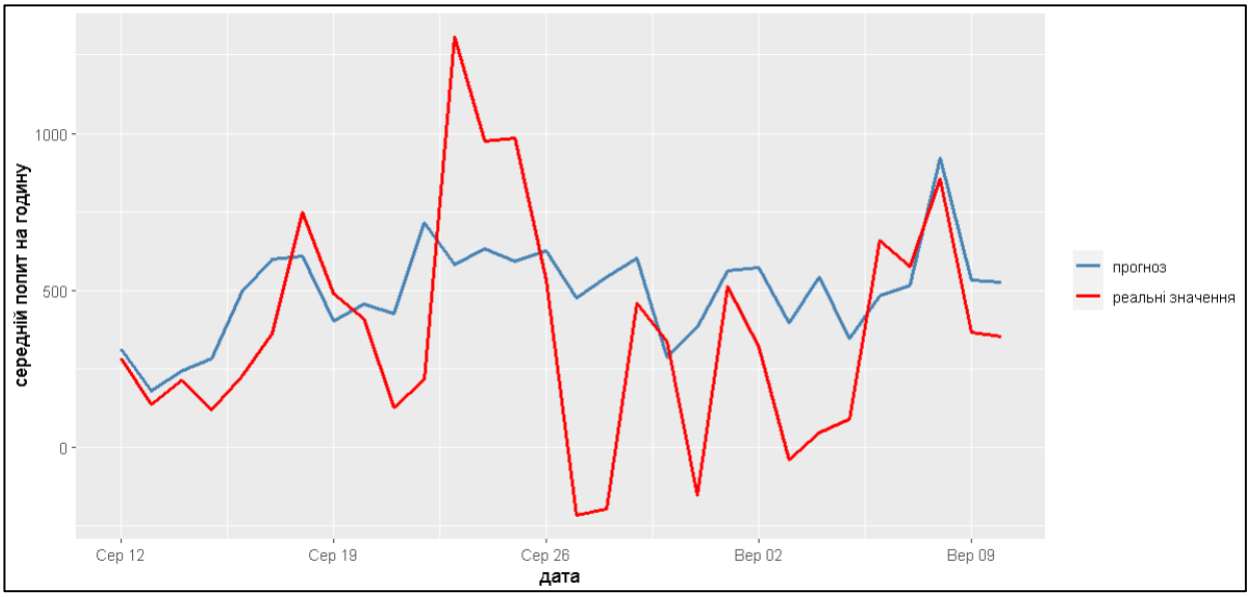

Рисунок 4.17 – Графік прогнозованого нейромережевою моделлю і реального попиту

4.5 Підсумки по окремим моделям

На рис. 4.18 наведено графік реальних і прогнозованих різними моделями значень попиту.

```
ggplot(data=test_set,aes(x=date, y=avg_demand ,colour="actual",linetype="ac
tual"))+
  geom_line(colour='black',lwd=1)+
  geom_line(data=prophet_forecast,aes(x=date,y=avg_demand,colour="prophet",
linetype="predicted"),lwd=1)+
  geom_line(data=ets_forecast,aes(x=date,y=avg_demand,colour="ets",linetype
="predicted"),lwd=1)+
  geom_line(data=arima_forecast,aes(x=date,y=avg_demand,colour="arima",line
type="predicted"),lwd=1)+
  geom_line(data=nn_forecast,aes(x=date,y=avg_demand,colour="nn",linetype="
predicted"),lwd=1)+
  labs(color = NULL,x="дата",y="середній попит на годину") +
  scale_color_manual(
    values=c(actual="black",prophet="steelblue",ets="limegreen",arima="oran
ge",nn="red"),
    limits=c("actual","prophet","ets","arima","nn"),
    labels=c("реальні дані","модель prophet","модель ETS","модель ARIMA","н
ейромережева модель")
  )+scale_linetype_manual(
    values=c(actual=3,predicted=1),
    labels=NULL
  )+guides(linetype="none")
```

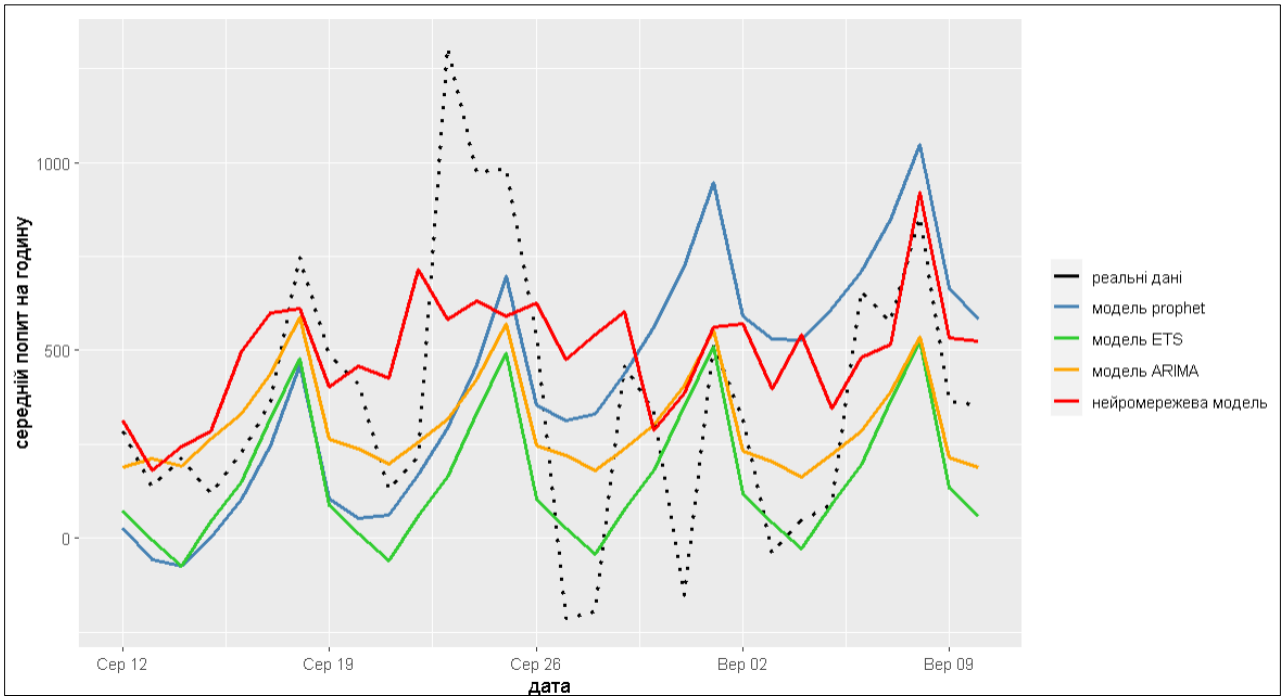

Рисунок 4.18 – Графік реальних і прогнозованих різними моделями значень попиту

На графіку можна помітити, що в різні проміжки часу різні моделі дають прогноз ближчий до реальних значень, і нема однієї моделі, яка б на всьому проміжку була б найближчою до реальних значень.

У табл. 4.1 підсумовано точність побудованих за різними методами окремих прогнозних моделей.

Таблиця 4.1 – Оцінки точності побудованих моделей

Модель	MSE	RMSE	MAE
GAM	157228,52	396,5205	324,3262
ETS	127475,51	357,0371	276,2925
ARIMA	93986,56	306,5723	228,7214
NNAR	110804,56	332,8732	256,0339

Таким чином найбільшу точність серед окремих моделей демонструє модель ARIMA.

4.6 Комбінування прогнозів

Створимо кілька комбінованих прогнозів, скориставшись функціями з пакету ForecastComb, і перевіримо їх точність.

```
library(ForecastComb)
fcomb <- foreccomb(
  observed_vector = train_set$avg_demand,
  prediction_matrix = as.matrix(data.frame(
    head( predict(prophet_model, make_future_dataframe(prophet_model,period
s=30))$yhat,803),
    ets_model$fitted,arima_model$fitted,nn_model$fitted)),
  newpreds = as.matrix(data.frame(
    prophet_forecast$avg_demand,ets_forecast$avg_demand,
    arima_forecast$avg_demand,nn_forecast$avg_demand)))
sa_comb <- comb_SA(fc)      #просте усереднення
median_comb <- comb_MED(fc) #медіана
bg_comb <- comb_BG(fc)      #метод мінімальної дисперсії
inv_comb <- comb_InvW(fc)   #метод оберненого рангу
ols_comb <- comb_OLS(fc)    #регресійна модель з коефіцієнтами
                          #підібраними методом найменших квадратів
lad_comb <- comb_LAD(fc)    #регресійна модель з коефіцієнтами
            #підібраними методом найменшого абсолютного відхилення
csr_comb <- comb_CSR(fc)    #комбінація кількох регресійних моделей
combined_forecasts_performance <- rbind(
  performanceMetrics(sa_comb$Forecasts_Test,test_set$avg_demand),
  performanceMetrics(median_comb$Forecasts_Test,test_set$avg_demand),
  performanceMetrics(bg_comb$Forecasts_Test,test_set$avg_demand),
  performanceMetrics(invw_comb$Forecasts_Test,test_set$avg_demand),
  performanceMetrics(ols_comb$Forecasts_Test,test_set$avg_demand),
  performanceMetrics(lad_comb$Forecasts_Test,test_set$avg_demand),
  performanceMetrics(csr_comb$Forecasts_Test,test_set$avg_demand))
rownames(combined_forecasts_performance) <- c("simple average","median","BG
","ols",'lad','inv.rank','csr')
```

```
> combined_forecasts_performance
                      mse      rmse      mae
simple average   91834.74 303.0425 214.0157
median           95734.16 309.4094 217.2909
BG               97225.87 311.8106 219.1354
ols              92097.26 303.4753 216.4003
lad             101700.17 318.9046 233.0618
inv.rank        103181.16 321.2182 239.1892
csr              97019.76 311.4799 219.2857
```

Рисунок 4.19 – Точність комбінованих прогнозів

Порівнявши отримані оцінки точності комбінованих прогнозів (рис. 4.19) можна зробити висновок, що декілька методів комбінування дозволили отримати точніший прогноз, проте просте усереднення прогнозів виявилося найбільш ефективним підходом, перевершивши складніші методи. Також воно є більш ефективним за підхід вибору найкращого окремого прогнозу, яким в даному випадку є модель ARIMA (табл. 4.1). На рис. 4.20 наведено графік прогнозів окремих моделей і комбінованого прогнозу.

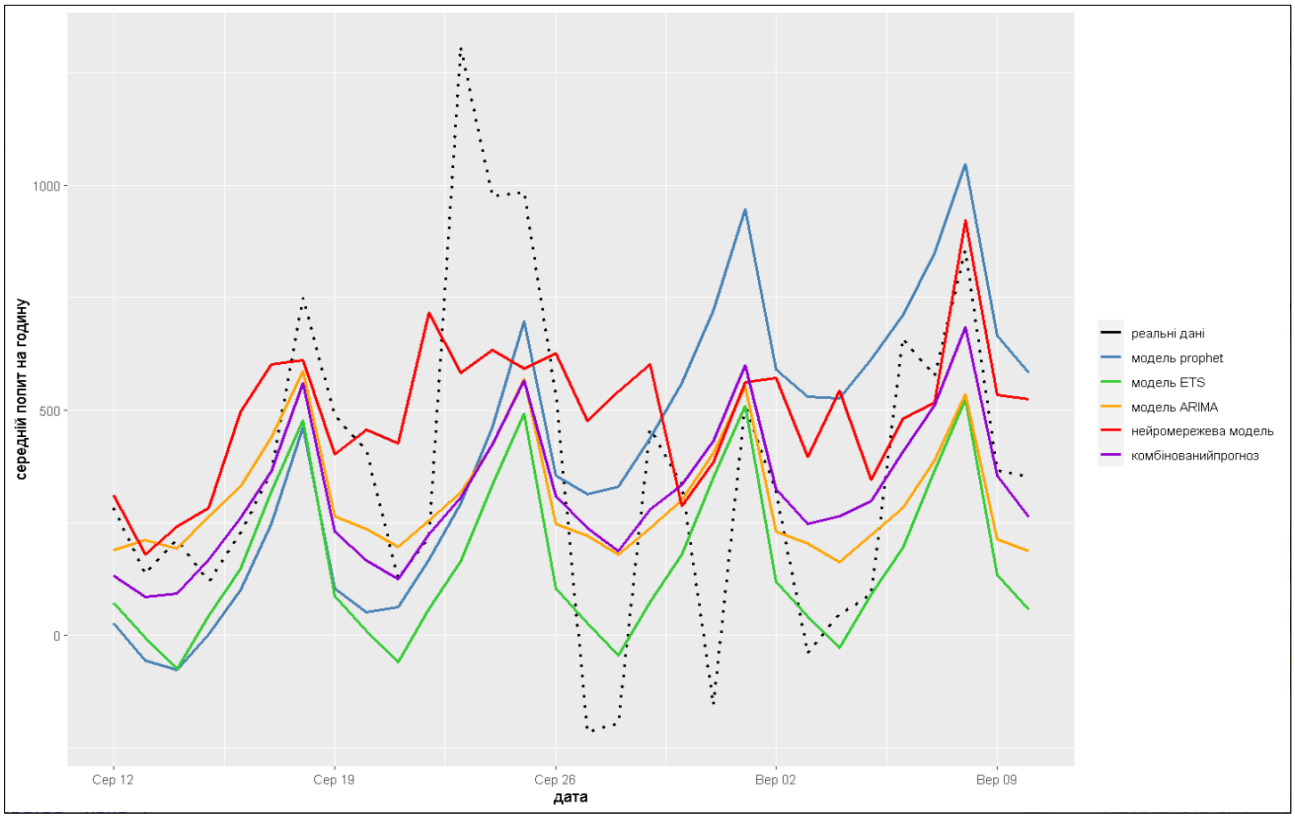

Рисунок 4.20 – Графік окремих прогнозів і комбінованого прогнозу

Висновки до розділу 4

В даному розділі було побудовано 4 прогнозні моделі: модель prophet, модель експоненційного згладжування, модель ARIMA і модель нейромережевої авторегресії. Для кожного типу моделі було побудовано кілька альтернативних моделей, оцінено їх точність на навчальній і тестовій вибірці і обрано оптимальну модель. За результатами моделювання найкращою окремою моделлю є модель ARIMA. Було виявлено, що найоптимальнішим підходом до комбінування моделей в даному випадку є просте усереднення прогнозів, що призводить до результату кращого за найкращу окрему модель, таким чином шляхом комбінування прогнозів вдалося підвищити точність прогнозу.

Лістинг коду моделювання, прогнозування, комбінування прогнозів і оцінки точності прогнозів наведено в додатку Б.

ВИСНОВКИ

Метою кваліфікаційної роботи було покращення точності прогнозування часового ряду шляхом використання методів комбінування прогнозів у застосуванні до задачі прогнозування попиту на електроенергію в енергомережі України.

Для виконання мети роботи було виділено задачі: аналіз теоретичного матеріалу і публікацій на тему, вибір набору даних, його аналіз і попередня обробка, моделювання і оцінка точності окремих прогнозних моделей і комбінованого прогнозу.

Було розглянуто існуючі методи прогнозування часових рядів на основі машинного навчання, а саме: узагальнена адитивна модель, модель експоненційного згладжування, модель ARIMA і модель нейромережевої авторегресії. Для кожного методу було побудовано кілька моделей, точність яких оцінено на навчальній і тестовій вибірці, і обрано оптимальну модель, таким чином було отримано 4 окремі моделі.

Було розглянуто існуючі методи комбінування прогнозів, проаналізовано публікації на тему комбінування прогнозів. Було застосовано декілька методів комбінування прогнозів для отримання комбінованих прогнозів із прогнозів окремих моделей. Декілька методів комбінування продемонстрували покращення точності прогнозу в порівнянні з найкращою окремою моделлю, серед них найвищу точність має метод простого усереднення, що підтверджує існування так званої «загадки комбінування прогнозів», часто згадуваної в публікаціях на цю тему.

Загалом використання комбінування прогнозів дозволило підвищити точність прогнозування часового ряду.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Hyndman, R., Athanasopoulos, G. Forecasting: principles and practice, 3rd edition, OTexts: Melbourne, Australia. 2021.
2. Hyndman, R.J., Athanasopoulos, G. Forecasting: principles and practice, 2nd edition, OTexts: Melbourne, Australia. 2018.
3. Hyndman, R.J., Athanasopoulos, G. Forecasting: principles and practice (3rd ed). OTexts : вебсайт. URL: https://otexts.com/fpp3/ (дата звернення: 20.08.2024).
4. А. Т. Яровий, Є. М. Страхов. Аналіз часових рядів. — Одеса : Освіта України, 2019.
5. Ravindra K., Rattan P., Mor S., Aggarwal A. N. Generalized additive models: Building evidence of air pollution, climate change and human health. Environment International. 2019. Vol. 132.
6. Understanding Generalized Additive Models (GAMs): A Comprehensive Guide. Analytics Vidhya : вебсайт. URL: https://www.analyticsvidhya.com/blog/2023/09/understanding-generalized-additive-models-gams-a-comprehensive-guide/ (дата звернення: 22.09.2024).
7. Taylor, S. J., Letham, B. (2018). Forecasting at scale. The American Statistician, 72(1), 37–45.
8. Hydman R., Koehler A., Keith J., Ralph D. Forecasting with Exponential Smoothing: The State Space Approach, Springer Science & Business Media : Berlin, Germany, 2008.
9. Ostertagova E., Ostertag O. (2012). Forecasting Using Simple Exponential Smoothing Method. Acta Electrotechnica et Informatica. 12. 62–66. 10.2478/v10198-012-0034-2.
10. Shumway, R., Stoffer, D. ARIMA Models. In: Time Series Analysis and Its Applications. Springer Texts in Statistics. Springer, Cham. (2017).

11. Introduction to ARIMA model. Medium : вебсайт. URL: https://medium.com/@ritusantra/introduction-to-arima-model-c8925103f4c7 (дата звернення: 20.08.2024).

12. Chapter 23: Using ARIMA for Time Series Analysis. A Language, not a Letter: Learning Statistics in R : вебсайт. URL: https://ademos.people.uic.edu/Chapter23.html (дата звернення: 20.08.2024).

13. Субботін С. О. Нейронні мережі : теорія та практика: навч. посіб. / С. О. Субботін. – Житомир : Вид. О. О. Євенок, 2020. – 184 с. ISBN 978-966-995-189-

14. Explained: Neural networks. MIT News : вебсайт. URL: https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414 (дата звернення: 20.08.2024).

15. Remus, William & O'Connor, Marcus. (2001). Neural Networks for Time-Series Forecasting. 10.1007/978-0-306-47630-3_12.

16. Winkler R., Makridakis S. (1983). The Combination of Forecasts. Journal of the Royal Statistical Society. Series A (General). 146. 150-157. 10.2307/2982011.

17. Hibon M., Evgeniou T. (2005). To combine or not to combine: Selecting among forecasts and their combinations. International Journal of Forecasting. 21. 15-24. 10.1016/j.ijforecast.2004.05.002.

18. Wang Xiaoqian, Hyndman Rob, Li Feng, Kang Yanfei. (2022). Forecast combinations: An over 50-year review. International Journal of Forecasting. 39. 10.1016/j.ijforecast.2022.11.005.

19. Mancuso A, Werner L. (2019). A comparative study on combinations of forecasts and their individual forecasts by means of simulated series. Acta Scientiarum. Technology. 41. 41452. 10.4025/actascitechnol.v41i1.41452.

20. Cang, Shuang & Yu, Hongnian. (2014). A combination selection algorithm on forecasting. European Journal of Operational Research. 234.

21. Lee, Tae-Hwy. (2011) Combining Forecasts with Many Predictors. Advances in Economic Forecasting, Matthew L. Higgins, ed. Kalamazoo, MI: W.E. Upjohn Institute for Employment Research, pp. 149-172.

22. Pike Tyler,Vazquez-Grande Francisco. (2020) Combining Forecasts: Can Machines Beat the Average? https://ssrn.com/abstract=3691117

23. David Soule. (2019) Forecast Combination with Multiple Models and Expert Correlations. VCU Scholars Compass.

24. Stock James, Watson Mark. (2004). Combination Forecasts of Output Growth in a Seven-Country Data Set. Journal of Forecasting. 23. 405-430. 10.1002/for.928.

25. Diebold Francis, Shin Minchul. (2018). Machine learning for regularized survey forecast combination: Partially-egalitarian LASSO and its derivatives. International Journal of Forecasting. 35. 10.1016/j.ijforecast.2018.09.006.

26. Claeskens Gerda, Magnus Jan, Vasnev Andrey, Wang Wendun. (2016). The forecast combination puzzle: A simple theoretical explanation. International Journal of Forecasting. 32. 754-762. 10.1016/j.ijforecast.2015.12.005.

27. Smith Jeremy, Wallis Kenneth. (2013). A Simple Explanation of the Forecast Combination Puzzle. Oxford Bulletin of Economics and Statistics. 71. 331-355. 10.1111/j.1468-0084.2008.00541.x.

28. Holtrop N. (2014). Finding Effective Weights to Combine Forecasts. Econometrie. http://hdl.handle.net/2105/16506.

29. Jose V., Winkler R. (2008). Simple robust averages of forecasts: Some empirical results. International Journal of Forecasting, Volume 24, Issue 1, p. 163-169 https://doi.org/10.1016/j.ijforecast.2007.06.001.

30. Bates J.M. , Granger C.W.J. (1969). The combination of forecasts. Operational Research Quarterly 20(4): 451-468

31. Wei, Xiaoqiao. (2009). Regression-based Forecast Combination Methods. Journal for Economic Forecasting. 12. 5-18.

32. Burnham Kenneth, Anderson David. (2004). Model Selection and Multimodel Inference. A Practical Information-theoretic Approach. 10.1007/978-0-387-22456-5_5.

33. Montero-Manso Pablo, Athanasopoulos George, Hyndman Rob, Talagala Thiyanga. (2019). FFORMA: Feature-based forecast model averaging. International Journal of Forecasting. 36. 10.1016/j.ijforecast.2019.02.011.

34. Introduction to tsibble. The Comprehensive R Archive Network : вебсайт. URL: https://cran.rstudio.com/web/packages/tsibble/vignettes/intro-tsibble.html (дата звернення: 22.08.2024).

35. Forecast package. Rdocumentation : вебсайт. URL: https://www.rdocumentation.org/packages/forecast/versions/8.23.0 (дата звернення: 22.08.2024).

36. ForecastComb: Forecast Combination Methods. . The Comprehensive R Archive Network : вебсайт. URL: https://cran.r-project.org/web/packages/ForecastComb/index.html (дата звернення: 22.08.2024).

37. Christoph E. Weiss (2018) Forecast Combinations in R using the ForecastComb Package. The R Journal. URL: https://journal.r-project.org/articles/RJ-2018-052/ (дата звернення: 22.08.2024).

38. Energy Map : вебсайт. URL: https://map.ua-energy.org/uk/resources/5a616fba-fbc9-4073-9532-9161592faca8/ (дата звернення: 09.09.2024).

39. Оператор ринку : вебсайт. URL: https://www.oree.com.ua/ (дата звернення: 09.09.2024).

40. Перша перемога на енергетичному фронті: енергосистема України стала частиною енергосистеми Європи. Офіційний вебпортал Верховної Ради України : вебсайт. URL: https://www.rada.gov.ua/news/razom/220582.html (дата звернення: 12.09.2024).

41. Мастицький С.Е. Аналіз часових рядів за допомогою R (2020) : вебсайт, URL: https://ranalytics.github.io/tsa-with-r/ (дата звернення: 06.10.2024).

ДОДАТОК А

Лістинг коду попередньої обробки часового ряду

```
# бібліотеки, необхідні для роботи
# бібліотеки, необхідні для роботи
library(readxl)     # зчитування даних з excel-файлів
library(tibble)     # робота з таблицями даних
library(dplyr)      # маніпуляції з даними
library(ggplot2)    # створення графіків
library(ggrepel)    # додавання тексту до графіків
library(lubridate)  # робота з датами та часом
library(tsibble)    # робота з часовими рядами
library(imputeTS)   # заповнення пропущених значень в часових рядах
library(feasts)     # дослідження та аналіз часових рядів
library(tidyverse)  # бібліотеки для роботи з даними
library(lmtest)     # статистичні тести
library(nonlinearTseries) # аналіз нелінійності часових рядів
library(forecast)   # побудова прогнозних моделей для часових рядів
library(tseries)    # аналіз часових рядів
library(prophet)    # побудова прогнозних моделей prophet
library(ForecastComb) #комбінування прогнозів

#завантаження даних
data <- read.csv("D:/electricity_demand.csv")

#аналіз структури набору даних
str(data)
summary(as.factor(data$energy_system))#перегляд значень змінної energy_system

#обробка дат і часу
data <- data %>% mutate(ymd = as.Date(date))
data <- data %>% mutate(datetime =
make_datetime(year(ymd),month(ymd),day(ymd),hour))

#агрегування попиту окремих підсистем енергомережі в один часовий ряд
energy_demand <- data.frame(datetime = unique(data$datetime), demand=0)
for(i in 1:length(energy_demand$datetime))
{
  energy_demand$demand[i] =
sum(data$demand[which(data$datetime==energy_demand$datetime[i])])
}
str(energy_demand) #структура отриманого часового ряду

#перетворення на формат даних tsibble
energy_demand <- as_tsibble(energy_demand,index=datetime)

#обробка пропущених значень
scan_gaps(energy_demand) #перевірка наявності пропущених значень
energy_demand <- fill_gaps(energy_demand) #заповнення пропусків значеннями NA
length(which(is.na(energy_demand))) #перевірка кількості значень NA
energy_demand <- energy_demand %>%
  mutate(demand = na_locf(demand)) #заповнення пропущених значень методом LOCF
scan_gaps(energy_demand) #повторна перевірка наявності пропущених значень і NA
length(which(is.na(energy_demand)))

#побудова графіка часового ряду
plot(x=energy_demand$datetime,y=energy_demand$demand, type="l",
     xlab="час", ylab="попит (МВт*год)",
```

```
     main="Попит на електроенергію в енергосистемі України",col="steelblue")

#отримання фрагменту часового ряду з 1 червня 2022
energy_demand <- energy_demand[which(energy_demand$datetime >
make_datetime(2022,6,1,0,0)),]

#побудова графіка фрагмента часового ряду
plot(x=energy_demand$datetime,y=energy_demand$demand, type="l",
     xlab="час", ylab="попит (МВт*год)",
     main="Попит на електроенергію в енергосистемі України",col="steelblue")

#отримання основних статистичних показників часового ряду
summary(energy_demand$demand)

#декомпозиція часового ряду
energy_demand %>%
  model(STL(demand ~ trend()+season(window=13),robust=TRUE)) %>%
  components() %>%
  autoplot()

#перевірка наявності автокореляції
acf(energy_demand$demand) #графік ACF
pacf(energy_demand$demand) #графік PACF
#тести на автокореляцію
model <- lm(energy_demand$demand ~ energy_demand$datetime)#модель лінійної
регресії
dwtest(model)#тест Дарбіна-Вотсона
bgtest(model)#тест Бройша-Ґотфрі

#тести на нелінійність
nonlinearityTest(energy_demand$demand)

#тести на стаціонарність (кількість необхідних диференціювань)
unitroot_ndiffs(energy_demand$demand) #кількість диференціювань
unitroot_nsdiffs(energy_demand$demand) #кількість сезонних диференціювань
#диференціювання часового ряду
energy_demand_diff <- diff(energy_demand$demand)
#графік диференційованого часового ряду
plot(x=energy_demand$datetime[2:length(energy_demand$datetime)],
     y=energy_demand_diff, type="l",
     xlab="час", ylab="diff(demand)",
     main="Диференційований часовий ряд",col="steelblue")

#агрегування за датою
energy_demand_daily <- energy_demand %>%
  group_by_key() %>%
  index_by(date = ~ as_date(.)) %>%
  summarize(avg_demand = mean(demand))

#побудова графіка агрегованого за датою часового ряду
plot(x=energy_demand_daily$date,y=energy_demand_daily$avg_demand, type="l",
     xlab="дата", ylab="середній за годину попит (МВт*год)",
     main="Попит на електроенергію в енергосистемі України",col="steelblue")

#отримання основних статистичних показників часового ряду
summary(energy_demand_daily$avg_demand)

#декомпозиція часового ряду
energy_demand_daily %>%
```

```
  model(STL(avg_demand ~ trend()+season(window=13),robust=TRUE)) %>%
  components() %>%
  autoplot()

#перевірка наявності автокореляції
acf(energy_demand_daily$avg_demand) #графік ACF
pacf(energy_demand_daily$avg_demand) #графік PACF
#тести на автокореляцію
model <- lm(energy_demand_daily$avg_demand ~ energy_demand_daily$date)#модель
лінійної регресії
dwtest(model)#тест Дарбіна-Вотсона
bgtest(model)#тест Бройша-Ґотфрі

#тести на нелінійність
nonlinearityTest(energy_demand_daily$avg_demand)

#тести на стаціонарність (кількість необхідних диференціювань)
unitroot_ndiffs(energy_demand_daily$avg_demand) #кількість диференціювань
unitroot_nsdiffs(energy_demand_daily$avg_demand) #кількість сезонних
диференціювань
#диференціювання часового ряду
energy_demand_daily_diff <- diff(energy_demand_daily$avg_demand)
#графік диференційованого часового ряду
plot(x=energy_demand_daily$date[2:length(energy_demand_daily$avg_demand)],
     y=energy_demand_daily_diff, type="l",
     xlab="час", ylab="diff(avg_demand)",
     main="Диференційований часовий ряд",col="steelblue")

#розбиття набору даних на навчальну і тестову вибірки
train_set <- head(energy_demand_daily,length(energy_demand_daily$avg_demand)-30)
test_set <- tail(energy_demand_daily,30)
length(train_set$avg_demand)
length(test_set$avg_demand)
```

ДОДАТОК Б

Лістинг коду моделювання і прогнозування

```
#функція для обчислення значень MSE, RMSE, MAE
performanceMetrics <- function(predicted,actual) {
  return (data.frame(
    mse=mean((predicted-actual)^2),
    rmse=sqrt(mean((predicted-actual)^2)),
    mae=mean(abs(predicted-actual))
  ))
}

#функція для побудови графіка прогнозованих і реальних значень
plotPredictAndActual <- function(predict,actual){
  ggplot(data=predict,aes(x=date, y=predicted,colour="predicted"))+

geom_ribbon(data=predict,aes(x=date,ymin=ymin80,ymax=ymax80),colour="lightsteelblu
e",fill="lightsteelblue",alpha=0.3)+
    geom_line(colour='steelblue',lwd=1)+
    geom_line(data=actual,aes(x=date,y=avg_demand,colour="actual"),lwd=1)+
    labs(color = NULL,x="дата",y="середній попит на годину") +
    scale_color_manual(
      values=c(predicted="steelblue",actual="red"),
      limits=c("predicted","actual"),
      labels=c("прогноз","реальні значення")
    )
}

#МОДЕЛЬ PROPHET
library(prophet)#підключення бібліотеки prophet

#приведення навчальних даних до формату data.frame
#(необхідно для створення моделей prophet)
train_set_prophet <- data.frame(ds=train_set$date,y=train_set$avg_demand)
test_set_prophet <- data.frame(ds=test_set$date,y=test_set$avg_demand)

#створення моделі і прогнозу зі стандартними параметрами
prophet_m1 <- prophet(train_set_prophet)
forecast_prophet_m1 <- predict(prophet_m1,
make_future_dataframe(prophet_m1,periods = 30))

#модель з більшою кількістю вузлових точок тренду
#(за замовчуванням changepoint.prior.scale = 0.05)
prophet_m2 <- prophet(train_set_prophet, changepoint.prior.scale = 0.2)
forecast_prophet_m2 <- predict(prophet_m2,
make_future_dataframe(prophet_m2,periods = 30))

#модель з меншою кількістю вузлових точок тренду
#(за замовчуванням changepoint.prior.scale = 0.05)
prophet_m3 <- prophet(train_set_prophet, changepoint.prior.scale = 0.02)
forecast_prophet_m3 <- predict(prophet_m3,
make_future_dataframe(prophet_m3,periods = 30))

#модель з більшим впливом сезонних компонент
#(за замовчуванням seasonality.prior.scale = 10)
prophet_m4 <- prophet(train_set_prophet, seasonality.prior.scale = 20)
```

```
forecast_prophet_m4 <- predict(prophet_m4,
make_future_dataframe(prophet_m4,periods = 30))

#модель з мультиплікативною сезонністю
prophet_m5 <- prophet(train_set_prophet, seasonality.mode = "multiplicative")
forecast_prophet_m5 <- predict(prophet_m5,
make_future_dataframe(prophet_m5,periods = 30))

#розрахунок MSE, RMSE і MAE і покриття методом перехресної перевірки
prophet_m1_performance <- performance_metrics(
  cross_validation(prophet_m1,horizon=30,unit="days"),
  metrics = c("mse","rmse","mae","coverage"), rolling_window = 1)
prophet_m2_performance <- performance_metrics(
  cross_validation(prophet_m2,horizon=30,unit="days"),
  metrics = c("mse","rmse","mae","coverage"), rolling_window = 1)
prophet_m3_performance <- performance_metrics(
  cross_validation(prophet_m3,horizon=30,unit="days"),
  metrics = c("mse","rmse","mae","coverage"), rolling_window = 1)
prophet_m4_performance <- performance_metrics(
  cross_validation(prophet_m4,horizon=30,unit="days"),
  metrics = c("mse","rmse","mae","coverage"), rolling_window = 1)
prophet_m5_performance <- performance_metrics(
  cross_validation(prophet_m5,horizon=30,unit="days"),
  metrics = c("mse","rmse","mae","coverage"), rolling_window = 1)
prophet_cross_validation_performance <- data.frame(
  Model=c("m1","m2","m3","m4","m5"),
  MSE=c(prophet_m1_performance$mse,prophet_m2_performance$mse,
        prophet_m3_performance$mse,prophet_m4_performance$mse,
        prophet_m5_performance$mse),
  RMSE=c(prophet_m1_performance$rmse,prophet_m2_performance$rmse,
        prophet_m3_performance$rmse,prophet_m4_performance$rmse,
        prophet_m5_performance$rmse),
  MAE=c(prophet_m1_performance$mae,prophet_m2_performance$mae,
        prophet_m3_performance$mae,prophet_m4_performance$mae,
        prophet_m5_performance$mae),
  coverage=c(prophet_m1_performance$coverage,prophet_m2_performance$coverage,
        prophet_m3_performance$coverage,prophet_m4_performance$coverage,
        prophet_m5_performance$coverage)
)

#розрахунок MSE, RMSE і MAE на тестовій вибірці
prophet_m1_performance=performanceMetrics(tail(forecast_prophet_m1$yhat,30),test_s
et$avg_demand)
prophet_m2_performance=performanceMetrics(tail(forecast_prophet_m2$yhat,30),test_s
et$avg_demand)
prophet_m3_performance=performanceMetrics(tail(forecast_prophet_m3$yhat,30),test_s
et$avg_demand)
prophet_m4_performance=performanceMetrics(tail(forecast_prophet_m4$yhat,30),test_s
et$avg_demand)
prophet_m5_performance=performanceMetrics(tail(forecast_prophet_m5$yhat,30),test_s
et$avg_demand)

prophet_test_set_performance <- data.frame(
  Model=c("m1","m2","m3","m4","m5"),
  MSE=c(prophet_m1_performance$mse,prophet_m2_performance$mse,
        prophet_m3_performance$mse,prophet_m4_performance$mse,
        prophet_m5_performance$mse),
  RMSE=c(prophet_m1_performance$rmse,prophet_m2_performance$rmse,
         prophet_m3_performance$rmse,prophet_m4_performance$rmse,
```

```
         prophet_m5_performance$rmse),
  MAE=c(prophet_m1_performance$mae,prophet_m2_performance$mae,
        prophet_m3_performance$mae,prophet_m4_performance$mae,
        prophet_m5_performance$mae)
)

#візуалізація прогнозованих і реальних значень
plot(prophet_m2, forecast_prophet_m2) #побудована модель і прогноз
#графік прогнозованих і реальних значень з тестової вибірки
plot(prophet_m2, forecast_prophet_m2,xlab="дата",ylab="середній попит на годину")+
  coord_cartesian(xlim = c(as.POSIXct("2024-08-13"), as.POSIXct("2024-09-
11")),ylim=c(-1000,2000))+
  geom_line(data = test_set_prophet,aes(as.POSIXct(ds), y,colour="actual"))+
  labs(color = NULL) +
  scale_color_manual(
    values=c(predicted="steelblue",actual="red"),
    limits=c("predicted","actual"),
    labels=c("прогноз","реальні значення")
  )

#МОДЕЛЬ ЕКСПОНЕНЦІЙНОГО ЗГЛАДЖУВАННЯ

train_set_ts <- as.ts(train_set)#перетворення навчальної вибірки на тип ts
#моделі
ses_model <- ses(train_set_ts, h=30) #просте експоненційне згладжування
holt_model <- holt(train_set_ts, h=30) #метод Хольта
hw_model <- hw(train_set_ts, h=30) #метод Хольта-Вінтерса
aaa_model <- predict(ets(train_set_ts,"AAA"), h=30) #метод Хольта-Вінтерса з
адитивними помилками

#оцінка точності моделей на навчальній вибірці
ets_models_accuracy <-
rbind(accuracy(ses_model),accuracy(holt_model),accuracy(hw_model),accuracy(aaa_mod
el))
rownames(ets_models_accuracy) <- c("ses","holt","hw","aaa")
ets_models_accuracy

#інформаційні критерії AIC, AICc і BIC
ets_information_criteria <- data.frame(
  Model=c("ses","holt","hw","aaa"),
  AIC=c(ses_model$model$aic, holt_model$model$aic,
hw_model$model$aic,aaa_model$model$aic),
  AICc=c(ses_model$model$aicc, holt_model$model$aicc,
hw_model$model$aicc,aaa_model$model$aicc),
  BIC=c(ses_model$model$bic, holt_model$model$bic,
hw_model$model$bic,aaa_model$model$bic)
)
ets_information_criteria

ets_models_performance <- rbind(
  performanceMetrics(ses_model$mean,test_set$avg_demand),
  performanceMetrics(holt_model$mean,test_set$avg_demand),
  performanceMetrics(hw_model$mean,test_set$avg_demand),
  performanceMetrics(aaa_model$mean,test_set$avg_demand)
)
rownames(ets_models_performance) <- c("ses","holt","hw","aaa")
ets_models_performance

aaa_predict <- data.frame(
```

```
  date=test_set$date,
  predicted=aaa_model$mean,
  ymax80=aaa_model$upper[,1],
  ymin80=aaa_model$lower[,1],
  ymax95=aaa_model$upper[,2],
  ymin95=aaa_model$lower[,2]
)
plotPredictAndActual(aaa_predict,test_set)

#МОДЕЛЬ ARIMA

arima_m1 <- forecast(auto.arima(train_set_ts),h=30)
arima_m2 <- forecast(Arima(train_set_ts,order=c(2,1,2),season=c(0,0,2)),h=30)
arima_m3 <- forecast(Arima(train_set_ts,order=c(1,1,2),season=c(2,0,2)),h=30)
arima_m4 <- forecast(Arima(train_set_ts,order=c(2,1,2),season=c(1,0,2)),h=30)
arima_m5 <- forecast(Arima(train_set_ts,order=c(2,1,2),season=c(2,1,2)),h=30)

#інформаційні критерії
arima_information_criteria <- data.frame(
  AIC=c(arima_m1$model$aic, arima_m2$model$aic, arima_m3$model$aic,
arima_m4$model$aic, arima_m5$model$aic),
  AICc=c(arima_m1$model$aicc, arima_m2$model$aicc, arima_m3$model$aicc,
arima_m4$model$aicc, arima_m5$model$aicc),
  BIC=c(arima_m1$model$bic, arima_m2$model$bic, arima_m3$model$bic,
arima_m4$model$bic, arima_m5$model$bic)
)


#точність моделі на навчальній вибірці
arima_models_accuracy <- rbind(
  accuracy(arima_m1),
  accuracy(arima_m2),
  accuracy(arima_m3),
  accuracy(arima_m4),
  accuracy(arima_m5)
)
rownames(arima_models_accuracy) <- c(as.character(arima_m1$model),

as.character(arima_m2$model),as.character(arima_m3$model),

as.character(arima_m4$model),as.character(arima_m5$model))
arima_models_accuracy

#точність моделі на тестовій вибірці
arima_models_performance <- rbind(
  performanceMetrics(arima_m1$mean,test_set$avg_demand),
  performanceMetrics(arima_m2$mean,test_set$avg_demand),
  performanceMetrics(arima_m3$mean,test_set$avg_demand),
  performanceMetrics(arima_m4$mean,test_set$avg_demand),
  performanceMetrics(arima_m5$mean,test_set$avg_demand)
)
rownames(arima_models_performance) <- c(as.character(arima_m1$model),

as.character(arima_m2$model),as.character(arima_m3$model),

as.character(arima_m4$model),as.character(arima_m5$model))
arima_models_performance

arima_m4_predict <- data.frame(
```

```
  date=test_set$date,
  predicted=arima_m4$mean,
  ymax80=arima_m4$upper[,1],
  ymin80=arima_m4$lower[,1],
  ymax95=arima_m4$upper[,2],
  ymin95=arima_m4$lower[,2]
)
plotPredictAndActual(arima_m4_predict,test_set)

#НЕЙРОМЕРЕЖЕВА МОДЕЛЬ

nn_m1 <- nnetar(train_set_ts)
nn_m2 <- nnetar(train_set_ts, p = 25, P = 25, MaxNWts=1500)
nn_m3 <- nnetar(train_set_ts, p = 50, P = 50, repeats = 50, MaxNWts = 5000)
nn_m4 <- nnetar(train_set_ts, p = 100, P = 100, MaxNWts = 20000)
nn_m5 <- nnetar(train_set_ts, p = 100, P = 100, repeats = 100, MaxNWts = 20000)

nn_models_performance <- rbind(
  performanceMetrics(forecast(nn_m1,h=30)$mean,test_set$avg_demand),
  performanceMetrics(forecast(nn_m2,h=30)$mean,test_set$avg_demand),
  performanceMetrics(forecast(nn_m3,h=30)$mean,test_set$avg_demand),
  performanceMetrics(forecast(nn_m4,h=30)$mean,test_set$avg_demand),
  performanceMetrics(forecast(nn_m5,h=30)$mean,test_set$avg_demand)
)
nn_models_performance

nn_forecast <- data.frame(date=test_set$date,predicted=forecast(nn_m3,h=30)$mean)
ggplot(data=nn_forecast,aes(x=date, y=predicted,colour="predicted"))+
  geom_line(colour='steelblue',lwd=1)+
  geom_line(data=test_set,aes(x=date,y=avg_demand,colour="actual"),lwd=1)+
  labs(color = NULL,x="дата",y="середній попит на годину") +
  scale_color_manual(
    values=c(predicted="steelblue",actual="red"),
    limits=c("predicted","actual"),
    labels=c("прогноз","реальні значення")
  )

#підсумок по окремим моделям

models_performance <- rbind(
  performanceMetrics(prophet_forecast$avg_demand,test_set$avg_demand),
  performanceMetrics(ets_forecast$avg_demand,test_set$avg_demand),
  performanceMetrics(arima_forecast$avg_demand,test_set$avg_demand),
  performanceMetrics(nn_forecast$avg_demand,test_set$avg_demand)
)
models_performance

#графіки побудованих моделей
ggplot(data=test_set,aes(x=date, y=avg_demand
,colour="actual",linetype="actual"))+
  geom_line(colour='black',lwd=1)+

geom_line(data=prophet_forecast,aes(x=date,y=avg_demand,colour="prophet",linetype=
"predicted"),lwd=1)+

geom_line(data=ets_forecast,aes(x=date,y=avg_demand,colour="ets",linetype="predict
ed"),lwd=1)+
```

```
geom_line(data=arima_forecast,aes(x=date,y=avg_demand,colour="arima",linetype="pre
dicted"),lwd=1)+

geom_line(data=nn_forecast,aes(x=date,y=avg_demand,colour="nn",linetype="predicted
"),lwd=1)+
  labs(color = NULL,x="дата",y="середній попит на годину") +
  scale_color_manual(

values=c(actual="black",prophet="steelblue",ets="limegreen",arima="orange",nn="red
"),
    limits=c("actual","prophet","ets","arima","nn"),
    labels=c("реальні дані","модель prophet","модель ETS","модель
ARIMA","нейромережева модель")
  )+
  scale_linetype_manual(
    values=c(actual=3,predicted=1),
    labels=NULL
  )+
  guides(linetype="none")

#КОМБІНУВАННЯ ПРОГНОЗІВ

#об'єкт foreccomb необхідний для створення комбінованих прогнозних моделей
fcomb <- foreccomb(
  observed_vector = train_set$avg_demand,
  prediction_matrix = as.matrix(data.frame(
    head( predict(prophet_model,
make_future_dataframe(prophet_model,periods=30))$yhat,803),
    ets_model$fitted,arima_model$fitted,nn_model$fitted)),
  newpreds = as.matrix(data.frame(
    prophet_forecast$avg_demand,ets_forecast$avg_demand,
    arima_forecast$avg_demand,nn_forecast$avg_demand))
)

sa_comb <- comb_SA(fc)      #просте усереднення
median_comb <- comb_MED(fc) #медіана
bg_comb <- comb_BG(fc)      #метод мінімальної дисперсії
inv_comb <- comb_InvW(fc)   #метод оберненого рангу
ols_comb <- comb_OLS(fc)    #регресійна модель з коефіцієнтами
                            #підібраними методом найменших квадратів
lad_comb <- comb_LAD(fc)    #регресійна модель з коефіцієнтами
                            #підібраними методом найменшого абсолютного відхилення
csr_comb <- comb_CSR(fc)    #комбінація кількох регресійних

#точність комбінованих прогнозів
combined_forecasts_performance <- rbind(
  performanceMetrics(sa_comb$Forecasts_Test,test_set$avg_demand),
  performanceMetrics(median_comb$Forecasts_Test,test_set$avg_demand),
  performanceMetrics(bg_comb$Forecasts_Test,test_set$avg_demand),
  performanceMetrics(invw_comb$Forecasts_Test,test_set$avg_demand),
  performanceMetrics(ols_comb$Forecasts_Test,test_set$avg_demand),
  performanceMetrics(lad_comb$Forecasts_Test,test_set$avg_demand),
  performanceMetrics(csr_comb$Forecasts_Test,test_set$avg_demand)
)

rownames(combined_forecasts_performance) <- c("simple
average","median","BG","ols",'lad','inv.rank','csr')
combined_forecasts_performance
```

```
#графік індивідуальних прогнозів і усередненого прогнозу
ggplot(data=test_set,aes(x=date, y=avg_demand
,colour="actual",linetype="actual"))+
  geom_line(colour='black',lwd=1)+

geom_line(data=prophet_forecast,aes(x=date,y=avg_demand,colour="prophet",linetype=
"predicted"),lwd=1)+

geom_line(data=ets_forecast,aes(x=date,y=avg_demand,colour="ets",linetype="predict
ed"),lwd=1)+

geom_line(data=arima_forecast,aes(x=date,y=avg_demand,colour="arima",linetype="pre
dicted"),lwd=1)+

geom_line(data=nn_forecast,aes(x=date,y=avg_demand,colour="nn",linetype="predicted
"),lwd=1)+

geom_line(data=sa_predict,aes(x=date,y=avg_demand,colour="sa",linetype="combined")
,lwd=1)+
  labs(color = NULL,x="дата",y="середній попит на годину") +
  scale_color_manual(

values=c(actual="black",prophet="steelblue",ets="limegreen",arima="orange",nn="red
",sa="darkviolet"),
    limits=c("actual","prophet","ets","arima","nn","sa"),
    labels=c("реальні дані","модель prophet","модель ETS","модель
ARIMA","нейромережева модель","комбінованийпрогноз")
  )+
  scale_linetype_manual(
    values=c(actual=3,predicted=1,combined=1),
    labels=NULL
  )+
  guides(linetype="none")
```

ДОДАТОК В

Апробація роботи

Робота пройшла апробацію під час Всеукраїнської науково-практичної конференції молодих вчених, аспірантів, студентів «Інтелектуальні інформаційні системи» 2-4 грудня 2024 р. у м. Миколаєві.

Міністерство освіти і науки України
Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Інформаційний лист

Всеукраїнська науково-практична конференція молодих вчених, аспірантів і студентів

Інтелектуальні інформаційні системи

2 – 4 грудня 2024 року

Миколаїв

УДК 004.8 + 519.226

Мельничук М. С., Калініна І. О.
Чорноморський національний університет ім. Петра Могили, Миколаїв, Україна

ІНТЕЛЕКТУАЛЬНА СИСТЕМА МОДЕЛЮВАННЯ ТА ПРОГНОЗУВАННЯ НА ОСНОВІ МЕТОДІВ КОМБІНУВАННЯ

Звичайним підходом при прогнозуванні часових рядів є побудова декількох прогнозних моделей за різними існуючими методами, серед яких обирають найточнішу модель. Проте проблемою такого підходу є втрата частини інформації, наявної в альтернативних моделях, що допомогла б покращити якість прогнозу. Альтернативним підходом для покращення якості прогнозних рішень є комбінування прогнозів. Об'єднання та комбінування кількох прогнозів, отриманих на основі одного набору даних, в даний час широко використовується для підвищення точності за рахунок інтеграції інформації, отриманої з різних джерел. Це знижує ризик визначення одного «найкращого» прогнозу. Існують різні методи комбінування прогнозів: від простого усереднення прогнозованих значень до складніших підходів з визначенням вагових коефіцієнтів прогнозних моделей. Ефективність методів комбінування прогнозів у порівнянні з вибором найкращого окремого прогнозу підтверджується у багатьох публікаціях на тему [1]. На основі дослідження методів комбінування прогнозів розроблено схему інформаційної системи прогнозування часових рядів, яка представлена на рисунку 1.

Рисунок 1 – Схема інформаційної системи прогнозування на основі методів комбінування прогнозів

В якості прикладу застосування прийомів комбінування прогнозів часових рядів розглянуто часовий ряд попиту на електроенергію в енергосистемі України [2], горизонт прогнозування - 30 днів.