**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**
**ІМЕНІ ТАРАСА ШЕВЧЕНКА**
Факультет комп'ютерних наук та кібернетики
Кафедра теорії та технології програмування

# Кваліфікаційна робота
**на здобуття ступеня магістра**

за спеціальністю 122 Комп'ютерні науки
на тему:

**Створення програмної системи для аналізу часових рядів**

Виконав студент 2-го курсу магістратури
Халецький Богдан Вадимович _____
(підпис)

Науковий керівник:
професор, доктор фіз.-мат. наук
Пашко Анатолій Олексійович _____
(підпис)

Засвідчую, що в цій роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Роботу розглянуто й допущено до захисту на засіданні кафедри теорії та технології програмування
«____» _____ 2021 р., протокол № ___
Завідувач кафедри
М. С. Нікітченко _____
(підпис)

Київ – 2021

## Реферат

**Актуальність:** Не дивлячись на велику кількість наукових робіт по дослідженню даної теми, що було створено впродовж останніх 50-ти років, задача гнучкого та універсального прогнозування часових рядів ще не вирішена. Кожного року ведеться активна розробка нових підходів для вирішення цієї задача, включаючи методи штучного інтелекту. Актуальною задачею є адаптація сучасних підходів до вирішення задач бізнесу, а саме рітейлерських мереж.

**Мета:** Створити апаратно-програмний комплекс для отримання інформації про стан полиць магазинів та на основі отриманої інформації створити систему прогнозування попиту на товари, що знаходяться на полиці.

Мета реалізується у наступних **завданнях**:

1. Проаналізувати методи теорії аналізу часових рядів;
2. Порівняти ефективність, ресурсоємність та універсальність різних моделей для аналізу;
3. Створити програмно-апаратний комплекс для отримання інформації про стан полиці на основі ммікроконтролерів ESP32.
4. Створити сервіс прогнозування часових рядів на основі отриманих даних.
5. Реалізувати мікро сервісну архітектуру та контейнеризацію для сервісу прогнозування часових рядів.

**Об’єктом роботи** є теорія часових рядів і методи прогнозування часових рядів

**Предметом роботи** є методичні підходи і практичне застосування методів моделювання часових

**Методи дослідження:** загальнонаукові методи дослідження, статистичного та компаративного аналізу, емпіричного пізнання, аналіз експертних оцінок, проектування.

**Інструменти дослідження та розробки:** python 3.8 (numpy, pandas, sklearn, tensorflow, fbprophet, statsmodels, aiohttp, asyncio, scypi) jupyter notebook, ArduinoIDE, C++, ESP_IDF, Redis, Docker, JavaScript(React Framework, React bootstrap), S3

**Сфера застосування:** експертна система застосовується для покращення бізнес процесів у сфері рітейлу на основі IOT камер, методів машинного зору та методів прогнозування часових рядів.

**Значимість роботи:** результатом застосування системи є збільшення таких метрик, як доступність на полиці товару та прогнозування попиту на основі обороту певних товарів за час проведення спостережень. Також використання системи допомагає економити час персоналу, що призводить до потенційного зменшення використання людського ресурсу, а прогнозування попиту до оптимізації логістики і зменшення втрат через псування продуктів.

**Наукова новизна отриманих результатів** визначається тим, що:

- адаптація методів моделювання часових рядів для вирішення прикладних задач;
- проаналізовано ефективність методів в залежності від таких параметрів, як розмір та якість початкових вхідних даних, можливість динамічної корекції прогнозу на основі нових даних.

**Рекомендації щодо використання результатів роботи:** результати дослідження можуть бути використанні, як методичний матеріал із

застосування методів моделювання часових рядів для вирішення прикладних проблем.

# Contents

Реферат 2

Умовні позначення 6

Вступ 7

РОЗДІЛ 1 ТЕОРЕТИЧНІ ВІДОМОСТІ ТА ОСНОВНІ МЕТОДИ АНАЛІЗУ ЧАСОВИХ РЯДІВ 8

1.1 Теоретичні відомості 8

1.2 Моделі породження часових рядів 9

1.3 Тренд і його аналіз 14

1.4 Автокореляція рівнів часового ряду 15

1.5 Загальний огляд багатокрокових моделей прогнозування часових рядів 17

РОЗДІЛ 2 Аналіз моделей прогнозування часових рядів 22

2.1 Авторегресійні моделі 22

2.2 Модель ковзкого середнього 23

2.3 Методи аналізу трендів 24

2.4 Модель ARIMA 29

2.5 Модель PROPHET 32

2.6 Методи моделювання часових рядів на основі машинного навчання 33

2.7 Порівняння ефективності прогнозування моделей 35

РОЗДІЛ 3 Апаратно-програмна реалізація системи 42

3.1 Архітектура та елементи системи 42

3.2 Реалізація системи камер 44

3.2.1 Створення технічних вимог для системи камер 44

3.2.2 Платформа ESP-32 45
3.2.3 Реалізація функціоналу системи камер на основі плати ESP32-cam 46
3.3 Реалізація системи прогнозування часових рядів 53
3.3.1 Створення технічних вимог для системи прогнозування часових рядів 53
3.3.2 Організація системи з точки зору програмної архітектури 55
ВИСНОВКИ 56
Література: 57

## Умовні позначення

AR — autoregressive. Авторегресійна;

MA — moving average. Ковзне середнє;

ARMA — Autoregressive moving average. Авторегресії ковзного середнього;

ARIMA — Autoregressive integrated moving average. Інтегрована авторегресія - ковзного середнього;

MAPE — mean absolute percentage error. Середня абсолютна похибка;

ARM — additive regression model. Адитивна регресійна модель;

LSTM – long short-term memory. Тривала короткочасна пам’ять;

OSA – on shelf availability. Процент заповненності полиці;

OOS – out of stock. Процент відсутнього товару.

# Вступ

Часовий ряд - це послідовність значень будь-якої величини в різні моменти часу. Ми зустрічаємо їх часто як у повсякденному житті, так і науковій діяльності. У медицині це можуть бути результати кардіограми; в геології це може бути відлуння (ехо), створене землетрусом; в фізиці це можуть бути послідовні значення величин, за якими ведеться спостереження, наприклад, кількість радіоактивних молекул, що пройшли через розпад у даний момент часу; у економіці це можуть бути зміни рівня безробіття або процентних ставок. Результати аналізу часових рядів використовується для вирішення таких важливих задач, як передбачення землетрусів, прогноз погоди, математичних фінансах, електроенцефалографія, інженерія управління, астрономія, інженерній комунікація, і загалом, в будь-якій галузі прикладної науки і техніки, яка передбачає тимчасові вимірювання.

Аналіз часових рядів на відміну від аналізу випадкових вибірок ґрунтується на припущенні, що послідовні значення даних спостерігаються через рівні проміжки часу. Тобто існують дві основних мети аналізу часових рядів: (1) визначення природи ряду і (2) прогнозування (передбачення майбутніх значень часового ряду по справжніх і минулих значеннях). Обидві ці цілі вимагають, аби модель ряду була ідентифікована і формально описана. Як тільки модель визначена, можливо з її допомогою інтерпретувати дані та використовувати у змістовній теорії для розуміння динаміки змін, а також екстраполювати потім ряд на основі знайденої моделі, тобто передбачити його майбутні значення.

# РОЗДІЛ 1 ТЕОРЕТИЧНІ ВІДОМОСТІ ТА ОСНОВНІ МЕТОДИ АНАЛІЗУ ЧАСОВИХ РЯДІВ

## 1.1 Теоретичні відомості

Аналіз часових рядів — сукупність математико-статистичних методів аналізу, призначених для виявлення структури часових рядів і для їх прогнозування. Сюди належать, зокрема, методи регресійного аналізу. Виявлення структури часового ряду необхідно для того, щоб побудувати математичну модель того явища, яке є джерелом аналізованого часового ряду. Прогноз майбутніх значень часового ряду використовується для ефективного прийняття рішень.

Існують дві основні мети аналізу часових рядів: визначення природи ряду і прогнозування (пророкування майбутніх значень часового ряду по теперішнім і минулим значенням). Обидві ці цілі вимагають, щоб модель ряду була ідентифікована і, більш-менш, формально описана. Як тільки модель визначена, ми можемо з її допомогою інтерпретувати представлені дані (використовувати у предметній області, наприклад, для розуміння сезонної зміни цін на товари, якщо займаємося економікою). Не звертаючи уваги на глибину розуміння і справедливість теорії, ми можемо екстраполювати потім ряд на основі знайденої моделі, тобто передбачити його майбутні значення.

## 1.2 Моделі породження часових рядів

**Статистичний підхід до аналізу часових рядів.**

Одним із основний підходів до аналізу часових рядів є статистичний підхід. Основна особливість статистичного аналізу часових рядів полягає в тому, що послідовність спостережень $X_1,\ldots, Xn$ розглядається як реалізація послідовності статистично залежних випадкових величин $X_1,\ldots, Xn$, що мають деякий спільний розподіл з функцією розподілу:

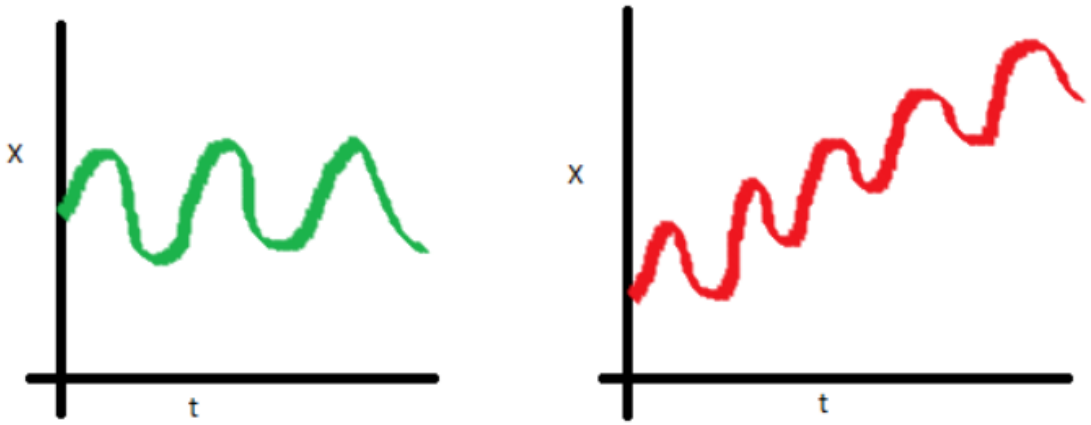
$$F(v1, v2, \ldots , vn) = P\{X_1< v1, X_2< v2,\ldots,Xn<vn\}.(1.1)$$

Розглянемо в основному часові ряди, в яких спільний розподіл випадкових величин $X_1,\ldots, Xn$ має спільну щільність розподілу $P(X_1, X2,\ldots,Xn)$.

Щоб зробити завдання статистичного аналізу часових рядів доступним для практичного застосування, доводиться так чи інакше обмежувати клас даних моделей часових рядів, вводячи ті або інші припущення відносно структури ряду і структури його імовірнісних характеристик. Одне з подібних обмежень передбачає таку властивість, як стаціонарність часового ряду.

Стаціонарний часовий ряд - це той, чиї статистичні властивості, такі як метематичне сподівання, дисперсія, автокореляція і т.д., є постійними протягом часу [1]. Більшість методів статистичного прогнозування ґрунтуються на припущенні, що часові ряди можуть бути надані приблизно стаціонарними (тобто "стаціонарними") за допомогою використання математичних перетворень. Стандартизовану серію порівняно легко

передбачити: ви просто передбачаєте, що її статистичні властивості будуть такими ж у майбутньому, як і в минулому. Наглядним прикладом є рис.1, на якому зображено два графіки. Перший є стаціонарним, а для другого математичне сподівання збільшується з часом.




«Рисунок 1.1 – Стаціонарний і нестаціонарний ряд»

Міра тісноти статистичного зв'язку між випадковими величинами Xt і Xt+1 може бути виміряна парним коефіцієнтом кореляції

$$Corr\left(X_t, X_{t+\tau}\right) = \frac{Cov\left(X_t, X_{t+\tau}\right)}{\sqrt{D\left(X_t\right)} \cdot \sqrt{D\left(X_{t+\tau}\right)}}$$

де:

$$Cov(X_t, X_{t+\tau}) = E[\left(X_t - E\left(X_t\right)\right)(X_{t+\tau} - E\left(X_{t+\tau}\right)].$$

Якщо ряд Xt стаціонарний, то значення $Cov(X_t, X_{t+\tau})$ не залежить від t і є функцією лише від τ ми використовуватимемо для нього позначення γ(τ):

$$\gamma(\tau) = Cov\left(X_t, X_{t+\tau}\right);$$

Зокрема,

$$D\left(X_t\right) = Cov\left(X_t, X_t\right) = \gamma(0).$$

Відповідно, для стаціонарного ряду і значення коефіцієнта кореляції $Corr(X_t, X_{t+\tau})$ залежить лише від t , ми використовуватимемо для нього позначення $p(\tau)$, отже

$$p(\tau) = Corr\left(X_t, X_{t+\tau}\right) = \frac{\gamma(\tau)}{\gamma(0)};$$

Зокрема p(0) = 1 .

Практична перевірка строгої стаціонарності ряду Xt на підставі спостереження значень X1,X2,..,Xn в загальному випадку досить складна.

У зв'язку з цим під стаціонарним рядом на практиці часто розуміють часовий ряд Xt, в якого

$$E\left(X_t\right) \equiv \mu,$$

$$D\left(X_t\right) \equiv \sigma^2,$$

$$Cov\left(X_t, X_{t+\tau}\right) \equiv \gamma(\tau) \text{ для будь } - \text{ яких } t, \tau.$$

Ряд, для якого виконано вказані три умови, називають стаціонарним в широкому сенсі (слабко стаціонарним, стаціонарним другого порядку або коваріаційно стаціонарним).

Якщо ряд є стаціонарним в широкому сенсі, то він не обов'язково є строго стаціонарним. В той же час, і строго стаціонарний ряд може не бути стаціонарним в широкому сенсі просто тому, що у нього можуть не існувати математичне сподівання і дисперсія. Крім того, можливі ситуації, коли вказані три умови виконуються, але, наприклад, E(Xt) залежить від t.

**Процесом білого шуму** називають стаціонарний часовий ряд $X_t$, для якого

$$E(X_t)\equiv 0,\ D(X_t)\equiv \sigma^2 > 0$$

і

$$p(\tau) = 0 \quad \text{якщо} \quad \tau \neq 0$$

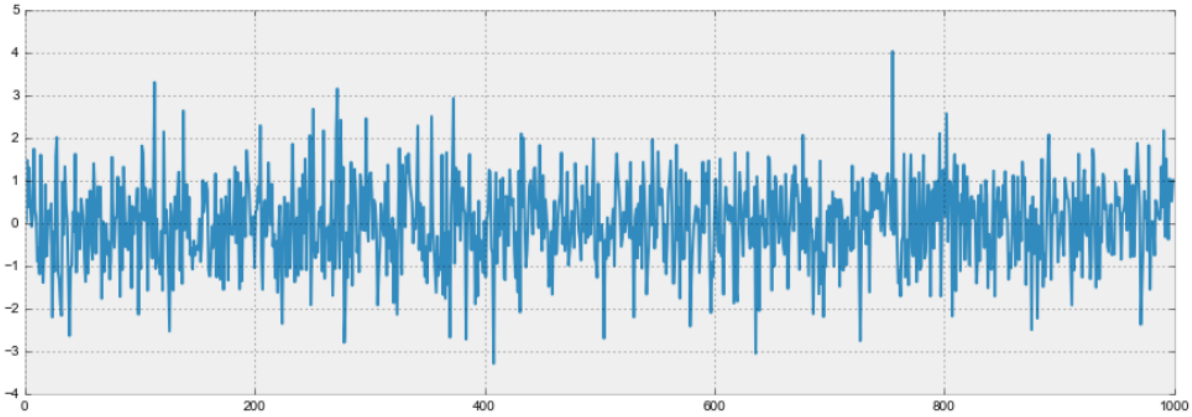
Останнє означає, що при t≠s випадкові величини Xt і Xs , відповідні спостереженням процесу білого шуму в моменти t і s, некорельовані.

У разі, коли Xt має нормальний розподіл, випадкові величини X1…, Xn взаємно незалежні і мають однаковий нормальний розподіл $N(0,\sigma^2)$, утворюючи випадкову вибірку з цього розподілу, тобто $Xt \sim i.i.d.\ N(0, \sigma^2)$.

Такий ряд називають білим шумом Гаусса. В той же час, в загальному випадку, навіть якщо деякі випадкові величини X1,..,Xn взаємно незалежні і мають однаковий розподіл, то це ще не означає, що вони утворюють процес

білого шуму, оскільки випадкова величина Xt може просто не мати математичного сподівання і дисперсії.

Часовий ряд, відповідний процесу білого шуму, поводиться нерегулярним чином через некорельованість при $t \neq s$ випадкових величин Xt і Xs .




«Рисунок 1.2 Білий шум згенерований автором»

Білий шум є важливою концепцією аналізу та прогнозування часових рядів.

Він є важливим з двох основних причин:

1. Передбачуваність: Якщо наш часовий ряд - це білий шум, то, за визначенням, він випадковий. Ми не можемо розумно моделювати його і робити прогнози.
2. Модельна діагностика: Серія помилок у моделі прогнозування часових рядів в ідеалі повинна бути білим шумом.

Модельна діагностика є важливою областю прогнозування часових рядів.

Очікується, що дані часових рядів містять деякі компоненти білого шуму поверх сигналу.

## 1.3 Тренд і його аналіз

Тренд або тенденція часового ряду - це деяке умовне поняття. Під трендом розуміють закономірну, невипадкову складову часового ряду (зазвичай монотонну), яка може бути обчислена за певним однозначним правилом. Тренд часового ряду часто пов'язаний з дією фізичних законів або яких-небудь інших об'єктивних закономірностей. Проте не можна однозначно розділити випадковий процес або часовий ряд на регулярну частину (тренд) і коливальну частину (залишок). Тому допускається, що тренд - це деяка функція простого вигляду (лінійна, квадратична, тощо ), що описує поведінку в цілому ряду або процесу. Якщо виділення такого тренду спрощує дослідження, тоді припущення про вибрану форму тренду вважається допустимим.

Для часового ряду рівняння лінійного тренду має вигляд:

$$x - E(X) = r\frac{\sigma_x}{\sigma_T}(t - E(T)).$$

При r>0 говорять про позитивний тренд (з часом значення часового ряду має тенденцію зростати), при r<0 про негативний (тенденція спадає). При r, близьких до нуля, інколи говорять про бічний тренд, для випадку, коли t=1,2,3...n, маємо:

$$E(X) = \frac{n+1}{2},\ D(T) = \frac{n(n+2)}{12},\qquad \text{а потім}\qquad \sigma_T = \sqrt{\frac{n(n+1)}{12}}$$

проте на практиці не варто окремо обчислювати r і $E(X)$ і лише потім підставляти їх в рівняння тренду. Краще прямо у формулі тренду провести спрощення, після яких вона набере вигляду:

$$x - E(X) = \frac{K_x}{D(T)\cdot\left(\frac{t-(n+1)}{2}\right)}.$$

Після виділення лінійного тренду потрібно з'ясувати, наскільки він вірогідний. Це робиться за допомогою аналізу коефіцієнта кореляції. Відмінність коефіцієнта кореляції від нуля і тим самим наявність реального тренду (позитивного або негативного) може виявитися випадковою, пов'язаною із специфікою даного відрізку часового ряду. Тобто, при аналізі іншого набору експериментальних даних (для того ж часового ряду) може виявитися, що отримана при цьому оцінка набагато ближче до нуля, ніж початкова (і, можливо, навіть має інший знак), і говорити про реальний тренд немає сенсу.

## 1.4 Автокореляція рівнів часового ряду

За наявності в часовому ряді тенденції і циклічних коливань значення кожного наступного рівня ряду залежать від попередніх. Кореляційну залежність між послідовними рівнями часового ряду називають автокореляцією рівнів ряду.

Кількісно її можна виміряти за допомогою лінійного коефіцієнта кореляції між рівнями вихідного часового ряду і рівнями цього ряду, зрушеними на декілька кроків в часі. Формула для розрахунку коефіцієнта автокореляції має вигляд:

$$r_1 = \frac{\sum_{t=2}^{n}(y_t - \bar{y}_1)(y_{t-1} - \bar{y}_2)}{\sqrt{\sum_{t=2}^{n}(y_t - \bar{y}_1)^2 \sum_{t=2}^{n}(y_{t-1} - \bar{y}_2)^2}},$$

де: $\underline{y_1} = \frac{1}{n-1}\sum_{t=2}^{n} y_t$, і $\underline{y_2} = \frac{1}{n-1}\sum_{t=2}^{n} y_{t-1}$.

Цю величину називають коефіцієнтом автокореляції рівнів ряду першого порядку, оскільки він вимірює залежність між сусідніми рівнями ряду.

Аналогічно можна визначити коефіцієнти автокореляції другого і вищих порядків. Так, коефіцієнт автокореляції другого порядку характеризує тісноту зв'язку між рівням і визначається по формулі:

$$r_2 = \frac{\sum_{t=3}^{n}(y_t - \bar{y}_3)(y_{t-2} - \bar{y}_4)}{\sqrt{\sum_{t=3}^{n}(y_t - \bar{y}_3)^2 \sum_{t=3}^{n}(y_{t-2} - \bar{y}_4)^2}},$$

де: $\underline{y_3} = \frac{1}{n-2}\sum_{t=2}^{n} y_t$, і $\underline{y_4} = \frac{1}{n-2}\sum_{t=2}^{n} y_{t-2}$.

Число періодів, за якими розраховується коефіцієнт автокореляції, називають лагом. Із збільшенням лага число пар значень, за якими розраховується коефіцієнт автокореляції, зменшується. Вважається за

доцільне для забезпечення статистичної достовірності коефіцієнтів автокореляції використовувати правило - максимальний лаг має бути не більше n/4 .

## 1.5 Загальний огляд багатокрокових моделей прогнозування часових рядів

Прогнозування часових рядів є зростаючою сферою інтересів, яка відіграє важливу роль майже в усіх сферах науки та техніки, таких як економіка, фінанси, метеорологія та телекомунікації [7]. На відміну від прогнозування на один крок вперед, задача з прогнозуванням на кілька кроків вперед є набагато більш складною [8], оскільки доводиться стикатися з різними додатковими ускладненнями такими, як накопичення помилок, зниження точності та підвищення невизначеності [9].

Тривалий час найбільш відомими і типовими підходами були так методи, як лінійні статистичні методи, наприклад, модель ARIMA, однак в кінці 1970-х і на початку 1980-х років стало все більш зрозумілим те, що лінійні моделі не пристосовані до широкого спектру реальних задач і є ефективними лише у вакуумі [10]. У той же самий період, було запропоновано кілька корисних нелінійних моделей часових рядів, таких як білійна модель [11], порогова авторегресивна модель [12] і авторегресивна умовна гетеросептична модель (ARCH) [13]. У наш час моделювання в Монте-Карло або методи завантаження використовуються для обчислення нелінійних прогнозів. Оскільки припущень не робиться щодо розподілу процесу помилок кращим є останній підхід [14]. Однак вивчення нелінійного аналізу часових рядів та прогнозування все ще знаходиться в зародковому стані порівняно з розвитком аналізу лінійних часових рядів [11].

В останні два десятиліття моделі машинного навчання привернули увагу світу та показали себе як серйозних претендентів на заміну класичних статистичних моделей. Ці моделі, які також називають black-box або моделі, керовані даними, є прикладами непараметричних нелінійних моделей, які використовують лише вхідні дані для вивчення стохастичної залежності між минулими і майбутніми значеннями рядів. Наприклад, Вербос встановив, що штучні нейронні мережі перевершують такі класичні статистичні методи, як лінійна регресія та підходи Бокса-Дженкінса з точки зору точності прогнозованих результатів, а також швидкості накопичення помилок, тобто можуть прогнозувати більшу кількість значень, не втрачаючи точність так швидко, як класичні моделі. Подібне дослідження було проведено Lapedes та Farber), які роблять висновок про те, що нейронні мережі можуть бути успішно використані для моделювання та прогнозування нелінійних часових рядів. Пізніше інших з'явилися такі моделі, як дерева рішень, векторні машини підтримки та регресія найближчих сусідів.

У сучасних дослідженнях звертають увагу на кілька аспектів процедури прогнозування, таких, як вибір моделі, ефект деасоналізації, поєднання прогнозів та багато інших. Однак підходам для створення багатоступеневих прогнозів. що базуються на машинному навчанні, не приділялося багато уваги через їх новизну.

Наскільки нам відомо, у літературі було запропоновано п'ять методів (або стратегій) для вирішення завдання прогнозування на X кроків уперед. Рекурсивна стратегія (Weigend та Gershenfenfeld, 1994; Sorjamaa et al., 2007; Cheng et al., 2006; Tao і Zhay, 1994; Klein, 2004; Hamzaebi та ін., 2009) , що базується на тому, що ми проходимо N разів на крок вперед, щоб отримати модель прогнозування для отримання N-того кроку прогнозу. Після оцінки

прогнозованих значень часового ряду, їх використовують для побудови наступного кроку моделювання.

На відміну від попередньої стратегії, яка використовує єдину модель, пряма (Direct) стратегія (Weigend та Gershenfeld 1994; Sorjamaa, 2007; Cheng, 2006; Tao і Zhay, 1994; Klein, 2004; Hamzaebi et al., 2009) оцінює набір H-моделей прогнозування, кожна з яких повертає прогноз для і-го значення (і ∈ {1, . H}).

У результаті поєднання двох попередніх підходів, був створений підхід DirRec (Direct recursive) (Sorjamaa and Lendasse, 2006). Ідея цього методу полягає в поєднанні аспектів обох аспектів прямої та рекурсивної стратегі. Іншими словами, на кожному кроці використовується інша модель, окрім того, результати попередніх кроків вводяться у вхідний набір даних для наступного кроку моделювання.

Щоб зберегти між прогнозованими значеннями стохастичну залежність, що характеризує часовий ряд, була введена стратегія "Multi-Input Multi-Output (MIMO)"(Bontempi, 2008; Bontempi and Ben Taieb, 2011; Kline, 2004). На відміну від попередніх стратегій, де моделі повертають скалярне значення, стратегія MIMO повертає вектор майбутніх значень за один крок вперед.

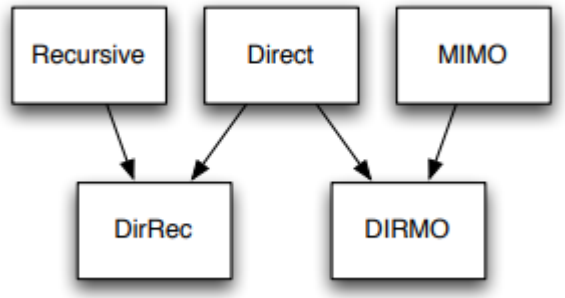
Остання стратегія, названа DIRMO (Ben Taieb et al., 2009), намагаються зберегти найпривабливіші аспекти стратегій DIRect та miMO. Ця стратегія спрямована на пошук компромісу між кращими: властивістю збереження стохастичної залежності між прогнозованими значеннями та гнучкістю процедури моделювання.

У літературі ці п'ять стратегій прогнозування представлені окремо, іноді, з використанням різних термінологій. Перший внесок щодо узагальнення - це ретельна уніфікація - огляд, а також теоретичний порівняльний аналіз існуючих стратегій на багато кроків вперед прогнозування, виконано у .

Незважаючи на те, що багато досліджень порівнювали багатокрокові підходи прогнозування часових рядів, спільнота не прийшла до єдиної щодо думки, який із підходів є найбільш ефективним. Тож у не можна однозначно виділити найбільш ефективний підхід, що дасть змогу відмовитися від усіх інших. Наприклад, дослідження (Bontempi., 1999; Weigend, 1992) дають експериментальні докази в користь рекурсивної стратегії проти прямої стратегії. Однак результати (Chjan та Hatchinson, 1994; Sorjamaa, 2007; Hamzaebi et al., 2009) підтримують той факт, що стратегія Direct краще, ніж рекурсивна стратегія. Робота (Sorjamaa and Lendasse, 2006) показує, що стратегія DirRec має кращу ефективність, ніж пряма(Direct) та рекурсивна стратегії. Пряма і рекурсивна стратегії були проаналізовані за допомогою теоретичного та емпіричного методів(Atiya et al., 1999). У результаті дослідження автори отримали явні докази на користь прямої стратегії(direct) .

Щодо стратегії MIMO, Kline (Kline, 2004) та Cheng (Cheng., 2006) висловили думку, що стратегія MIMO забезпечує гірші показники прогнозування, ніж рекурсивні та прямі стратегії. Однак у (Bontempi, 2008; Bontempi та Ben Taieb, 2011) порівнюючи MIMO, рекурсивний та прямий методи виступали за MIMO. Нарешті, (Ben Taieb et al., 2009, 2010) показує, що стратегія DIRMO дає кращі результати прогнозування, ніж прямі та MIMO стратегії, коли параметр, що контролює ступінь залежності між прогнозами, правильно ідентифікований. Ці порівняння проводилися з

різними наборами даних у різних конфігураціях з використанням різних методів прогнозування, таких як множинна лінійна регресія, штучні нейронні мережі, Приховані Маркові моделі та метод найближчих сусідів.




Кожна із представлених стратегій прогнозування вимагає визначення конкретної моделі прогнозування або алгоритму навчання для оцінки скалярної функції f або векторної функції F, які представляють часові стохастичні залежності. Розуміння особливостей багатоступеневих стратегій прогнозування дає можливість вибору для налаштування основної локальної моделі навчання для прогнозування експериментів кожної окремої проблеми.

# РОЗДІЛ 2 Аналіз моделей прогнозування часових рядів

## 2.1 Авторегресійні моделі

Часовий ряд - це послідовність вимірювань однієї і тієї ж змінної, виконаних у часі. Зазвичай вимірювання проводяться в рівномірно розподілені часи - наприклад, щомісячно або щорічно. Розглянемо спочатку проблему, в якій ми маємо y-змінну, виміряну як часовий ряд. Як приклад, ми можемо мати y як міру глобальної температури, при цьому вимірювання спостерігаються щороку. Щоб підкреслити, що ми виміряли значення протягом часу, ми використовуємо "t" як індекс, а не звичайний "i", тобто $y_t$ означає y, виміряний в періоді часу t. Модель авторегресії полягає в тому, що значення з часового ряду регресується на попередніх значеннях з того ж часового ряду. наприклад, $y_t$ через $y_{t-1}$:

$$y_t = \beta_0 + \beta_1 y_{t-1} + \varepsilon_t$$

У цій регресійній моделі змінна відповіді в попередньому періоді часу стала предиктором, і помилки мають звичайні припущення про помилки як в простій моделі лінійної регресії. Порядок авторегресії - це кількість безпосередньо попередніх значень у рядах, які використовуються для прогнозування значення в даний час. Отже, попередня модель - це авторегресія першого порядку, записана як AR(1).

Якщо ми хочемо передбачити у цього року $(y_t)$, використовуючи вимірювання глобальної температури в попередні два роки $(y_{t-1}, y_{t-2})$, то авторегресивна модель для цього буде:

$$y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \varepsilon_t$$

Ця модель є авторегресією другого порядку, записаною як AR (2), оскільки значення в момент часу t прогнозується з значень в моменти t − 1 і t − 2. Більш загально, авторегресія k-го порядку, записана як AR (k), є множинною лінійною регресією, в якій значення ряду в будь-який час t є (лінійною) функцією значень в часи t − 1, t − 2. ,…, T − k.

Зауважимо, що процес авторегресії буде стабільним, лише якщо параметри знаходяться в межах певного діапазону; наприклад, якщо є тільки один параметр авторегресії, то він повинен потрапляти в інтервал -1 < β <1. В іншому випадку попередні ефекти накопичуватимуться і значення послідовних $x_t$ будуть прямувати до нескінченності, тобто ряд не бути стаціонарним. Якщо є більш ніж один параметр авторегресії, то можуть бути визначені подібні (загальні) обмеження на значення параметрів [4].

Основна проблема моделей, що використовують суто авторегресійний підхід, складається у тому, що вони швидко накопичують помилки із кожним наступним прогнозованим наперед значенням. Одним із методів підлаштування коефіцієнтів є метод найменших квадратів (МНК).

## 2.2 Модель ковзкого середнього

В 1938 г. Вольд (Wold) довів наступний фундаментальний результат: будь-який слабкий стаціонарний часовий ряд може бути представлений у вигляді лінійної комбінації білих шумів, з різними ваговими коефіцієнтами.

Випадковий процес y (t) називається процесом змінного середнього MA (q) порядку q, якщо в розкладанні Вольда присутнє скіненне число доданків:

$$y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_n \varepsilon_1$$

де ε - це процес білого шуму у широкому змісті.

Процес білого шуму формально можна вважати процесом змінного середнього нульового порядку - MA (0).

Найчастіше на практиці використовують процес змінного середнього першого порядку MA (1):

$$y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1}$$

Назва «ковзне середнє» пояснюється тим, що поточне значення випадкового процесу визначається зваженим середнім q попередніх значень білого шуму. Процедуру змінного середнього часто використовують для того, щоб згладити дані, які сильно коливаються. Очевидно, що MA (q) - стаціонарний процес,

Для оцінки параметрів MA-моделей застосування звичайного МНК ускладнено, так як сума квадратів залишків не виражає аналітично через значення ряду. Можна використовувати метод максимальної правдоподібності в припущенні нормальності розподілу. Коваріаційна матриця необхідна для оцінки знаходиться виходячи з вищенаведених формул для коваріацій MA-процесу. Далі використовуються чисельні методи максимізації логарифмічної функції правдоподібності.

## 2.3 Методи аналізу трендів

Не існує перевірених "автоматичних" методів для визначення компонентів тренду у даних часових рядів; однак, поки тенденція є одноманітною (постійно збільшується або зменшується), ця частина аналізу даних, як правило, не дуже складна. Якщо дані часових рядів містять значну помилку, то перший крок у процесі ідентифікації тренда є згладжуванням.

Згладжування. Згладжування завжди передбачає певну форму локального усереднення даних таким чином, що несистематичні компоненти

індивідуальних спостережень виключають один одного. Найбільш поширеною методикою є згладжування ковзного середнього, яке замінює кожен елемент ряду або простою, або зваженою середньою з n оточуючих елементів, де n - ширина згладжуючого "вікна" [6]. Медіани можна використовувати замість засобів. Основною перевагою медіани в порівнянні зі згладжуванням ковзних середніх є те, що її результати є менш упередженими викидами (в межах вікна згладжування). Таким чином, якщо існують викиди в даних (наприклад, внаслідок помилок вимірювання), медіанне згладжування зазвичай створює більш гладкі або, щонайменше, більш "надійні" криві, ніж ковзний середній, заснований на тій же ширині вікна. Основним недоліком медіанного згладжування є те, що за відсутності чітких вирівнювань він може виробляти більше "зубчастих" кривих, ніж ковзне середнє, і це не дозволяє зважувати.

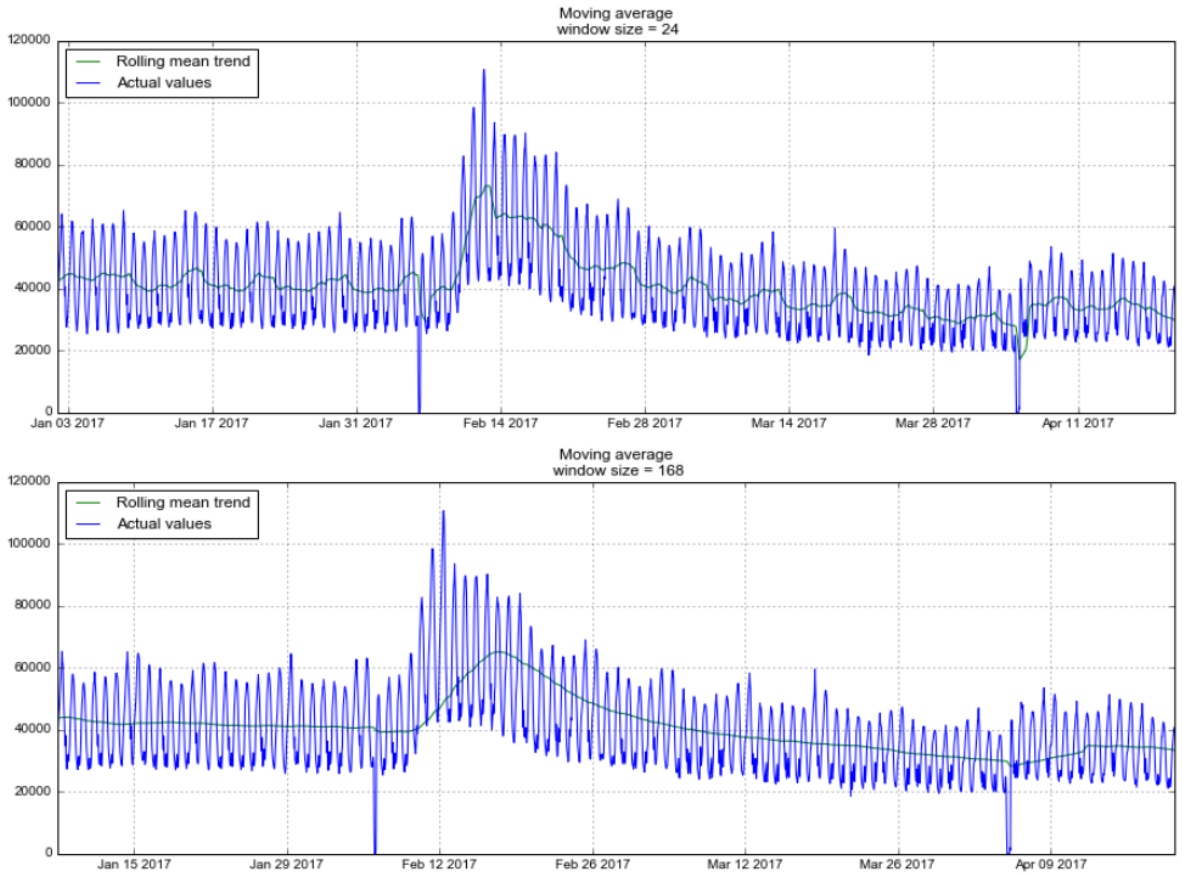
Метод рухомого середнього полягає в заміні фактичних значень членів ряду середнім арифметичним значень декількох найближчих до нього членів. Набір усереднених знаків так зване вікно. Член, значення якого замінюється на середню по вікну, займає в вікні серединне положення. Розрізняють два різновиди методу ковзкого середнього - просте згладжування і зважене згладжування.

Просте згладжування:

$$y_t = \frac{1}{n}\sum_{n=1}^{n-1} y_{t-n}$$

Модифікацією простого методу ковзкого середнього є зважене середнє, всередині якого спостереженням надаються різні ваги, що в сумі дають одиницю, при цьому зазвичай останнім спостереженням присвоюється більша вага.

$$y_t = \sum_{n=1}^{n-1} w_n y_{t+1-n}$$




«Рисунок 2.1 Метод ковзкого середнього з різним вікном»

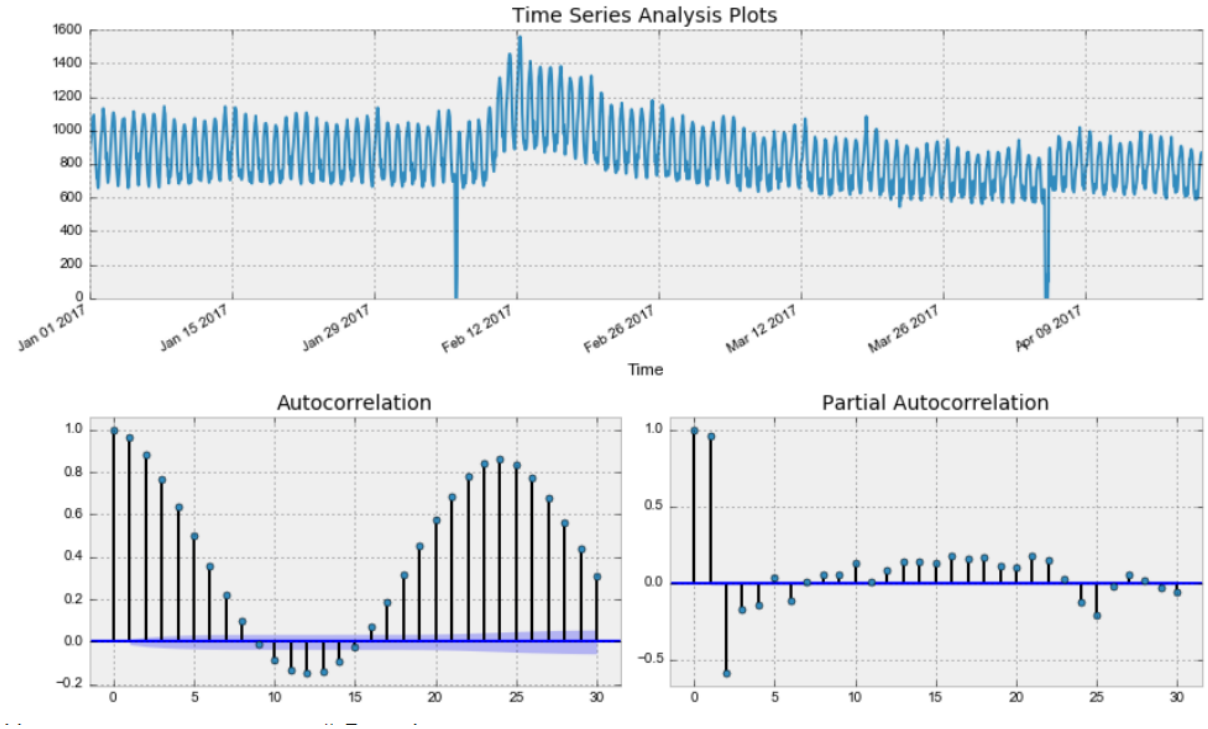
Автокорреляціонна корелограма. Сезонні тренди часових рядів можна вивчати за допомогою корелограм. Корелограма (autocorrelogram) відображає графічно та чисельно автокореляційну функцію (АКФ), тобто послідовні коефіцієнти кореляції (та їх стандартні помилки) для послідовних лагів у заданому діапазоні лагів (наприклад, від 1 до 30). Діапазони двох стандартних помилок для кожного відставання, як правило, відзначаються в корелограмах, але зазвичай розмір автокореляції представляє більший

інтерес, ніж його надійність, тому що ми зазвичай зацікавлені тільки в дуже сильних (і, отже, дуже значних) автокореляціях.

Ще одним із методів згладжування є експоненційне. Експоненці́йне згла́джування — це такий метод згладжування, що при кожній наступній ітерації враховуються всі попередні значення ряду, але ступінь врахування зменшується за експоненційним законом.

Вивчення корелограм. Під час вивчення корелограм необхідно пам'ятати, що автокореляції для послідовних лагів є формально залежними. Розглянемо наступний приклад. Якщо перший елемент тісно пов'язаний з другим, а другий - з третім, то перший елемент також повинен бути дещо пов'язаний з третім, і т.д. Це означає, що шаблон послідовних залежностей може значно змінитися після видалення першого в порядку автокореляції (тобто після диференціювання ряду з відставанням 1).

Часткові автокореляції. Іншим корисним методом для вивчення послідовних залежностей є дослідження часткової автокореляційної функції - розширення автокореляції, де залежність від проміжних елементів (тих, що знаходяться в межах відставання) видаляється. Іншими словами, часткова автокореляція аналогічна автокореляції, за винятком того, що при її розрахунку (авто) кореляції з усіма елементами в межах відстані розлучаються. Якщо визначено відставання 1 (тобто, в межах відставання немає проміжних елементів), то часткова автокореляція еквівалентна автокореляції. У певному сенсі часткова автокореляція забезпечує «чистіше» зображення послідовних залежностей для окремих лагів (не змішаних іншими послідовностями).




«Рисунок 2.2 Автокорелограма і часткова корелограма для нестаціонарного часового ряду»

Видалення послідовної залежності. Послідовна залежність для конкретного відставання k може бути видалена шляхом диференціювання ряду, тобто перетворення кожного i-го елемента ряду в його відмінність від (i-k) '-го елемента. Існує дві основні причини таких перетворень.

По-перше, можна виділити прихований характер сезонних залежностей у серії. Автокореляції для послідовних лагів є взаємозалежними. Таким чином, вилучення деяких автокореляцій змінить інші автокореляції, тобто може усунути їх або це може зробити деякі інші сезонність більш очевидними.

Інша причина усунення сезонних залежностей полягає в тому, щоб зробити серію стаціонарною, яка необхідна для ARIMA та інших методів.

## 2.4 Модель ARIMA

Методологія ARIMA, розроблена Боксом та Дженкінсом (1976), набула величезної популярності в багатьох областях, а дослідницька практика підтверджує її силу і гнучкість. Однак, завдяки своїй силі та гнучкості, ARIMA є складною технікою; вона не проста у використанні, вона вимагає великого досвіду, і хоча часто дає задовільні результати, ці результати залежать від рівня досвіду дослідника.

**Модель авторегресії ковзної середньої**. Загальна модель, введена Бокса Дженкінса включає в себе параметри авторегресії, а також ковзні середні, і явно включає диференціацію у формулюванні моделі. Зокрема, три типи параметрів у моделі: параметри авторегресії (p), кількість переходів диференціації (d) і параметри ковзних середніх (q). У позначеннях, введених Box та Jenkins, моделі узагальнюються як ARIMA (p, d, q); так, наприклад, модель, описана як (0, 1, 2), означає, що вона містить 0 (нуль) параметрів авторегресії (p) і 2 параметри ковзної середньої (q), які були обчислені для ряду після того, як він був один раз диференційований.

**Ідентифікація**. Як згадувалося раніше, вхідний ряд для ARIMA повинен бути стаціонарним, тобто він повинен мати постійне середнє значення, дисперсію і автокореляцію через час. Тому, як правило, перші рядки повинні бути диференційовані до тих пір, поки вони не стаціонарні (це також часто вимагає перетворення даних для стабілізації дисперсії). Кількість разів, необхідних для диференціації, щоб досягти стаціонарності, відображається в параметрі Для того, щоб визначити необхідний рівень диференціювання, слід вивчити ділянку даних і автокоррелограмму. Значні зміни в рівні (сильні зміни вгору або вниз) зазвичай вимагають відмінності першого порядку (відставання = 1); Сильні зміни нахилу зазвичай вимагають

несезонного диференціації другого порядку. Сезонні моделі вимагають відповідних сезонних різниць (див. Нижче). Якщо оцінені коефіцієнти автокореляції повільно зменшуються при більш довгих відставаннях, зазвичай необхідне диференціювання першого порядку. Проте слід пам'ятати, що деякі часові ряди можуть вимагати мало або взагалі не відрізняються, і що над диференційними рядами виробляються менш стійкі оцінки коефіцієнтів.

На цьому етапі необхідно також визначити, скільки параметрів авторегресії (p) і ковзних середніх (q) необхідно для отримання ефективної, але все ж таки охарактеризованої моделі процесу (песимічний, що він має найменшу кількість параметрів і найбільшу кількість ступенів свободи серед усіх моделей, які відповідають даним). На практиці числа параметрів p або q дуже рідко повинні бути більше 2.

**Оцінка та прогнозування**. На наступному етапі оцінюються параметри (з використанням процедур мінімізації функцій), так що сума квадратних залишків мінімізується. Оцінки параметрів використовуються на останньому етапі (прогнозування) для розрахунку нових значень ряду (крім включених у набір вихідних даних) та довірчих інтервалів для цих прогнозованих значень. Процес оцінки виконується на трансформованих (диференційованих) даних; перед тим, як будуть сформовані прогнози, серія повинна бути інтегрована (інтеграція є зворотною різницею), так що прогнози виражаються у значеннях, сумісних з вхідними даними. Ця функція автоматичної інтеграції представлена буквою I в назві методології (ARIMA = Auto-Regressive Integrated Moving Average).

**Константа в моделях ARIMA**. Крім стандартних авторегресійних і ковзних середніх параметрів, моделі ARIMA можуть також включати в себе

константу. Інтерпретація (статистично значущої) константи залежить від відповідної моделі. Зокрема, (1) якщо в моделі відсутні параметри авторегресії, то очікуваним значенням константи є середнє значення ряду; (2) якщо в серіях є авторегресивні параметри, то константа являє собою перехоплення. Якщо серія є диференційованою, то константа являє собою середнє значення або перехоплення ряду диференційованих рядів; наприклад, якщо ряд розрізняється один раз, а в моделі відсутні параметри авторегресії, то константа являє собою середнє значення диференційованих рядів, а отже, і нахил лінійного тренду недиференційованих рядів.

Існує кілька різних методів оцінки параметрів. Всі вони повинні видавати дуже схожі оцінки, але можуть бути більш-менш ефективними для будь-якої моделі. Загалом, під час фази оцінки параметрів використовується алгоритм мінімізації функцій для максимізації ймовірності спостережуваного ряду з урахуванням значень параметрів. На практиці це вимагає розрахунку (умовних) сум квадратів залишків з урахуванням відповідних параметрів. Запропоновано різні методи для обчислення сум квадратів для залишків: приблизний метод максимальної правдоподібності, метод точно-максимальної правдоподібності Мерольда.

**Порівняння методів.** Загалом, всі методи повинні давати дуже схожі оцінки параметрів. Крім того, всі методи мають однакову ефективність у більшості реальних додатків часових рядів. Тим не менш, спосіб 1 вище (приблизна максимальна ймовірність, відсутність зворотного зв'язку) є найшвидшим і повинен використовуватися, зокрема, для дуже довгих часових рядів (наприклад, з більш ніж 30000 спостережень). Метод точної максимальної правдоподібності Мерольда також може стати неефективним при використанні для оцінки параметрів для сезонних моделей з довгими сезонними відставаннями (наприклад, з річними відставаннями 365 днів). З

іншого боку, завжди слід спочатку використовувати приблизний метод максимальної правдоподібності, щоб встановити початкові оцінки параметрів, які дуже близькі до фактичних кінцевих значень; таким чином, зазвичай тільки кілька ітерацій з точним методом максимальної правдоподібності (3, вище) необхідні для завершення оцінки параметрів.

**Параметри стандартних помилок**. Для всіх оцінок параметрів буде обчислено так звані асимптотичні стандартні помилки. Вони обчислюються з матриці часткових похідних другого порядку, що апроксимується через кінцеве диференціювання.

**Оцінка помилки.** Як зазначалося вище, процедура оцінювання вимагає мінімізувати (умовні) суми квадратів залишків ARIMA. Якщо модель неприйнятна, то в процесі ітеративної оцінки може статися, що оцінки параметрів стають дуже великими і, по суті, недійсними. У такому випадку вона призначатиме сумам квадратів дуже велике значення (так зване штрафне значення). Зазвичай це "спонукає" ітераційний процес переміщення параметрів з недійсних діапазонів. Проте в деяких випадках навіть ця стратегія виходить з ладу, і ми можемо побачити на екрані (під час процедури оцінки) дуже великі значення для SS в послідовних ітераціях. У цьому випадку потрібно ретельно оцінити відповідність нашої моделі. Якщо модель містить багато параметрів і, можливо, компонент втручання, ми можемо спробувати знову з різними початковими значеннями параметрів.

## 2.5 Модель PROPHET

Дана модель була розроблена компанією FACEBOOK для створення якісних прогнозів, без попередніх дій. Ця модель складається з наступних компонентів:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t$$

Сезонні компоненти $s(t)$ відповідають за моделювання періодичних змін, пов'язаних з тижневою та річною сезонністю. Тижнева сезонність моделюється за допомогою примітивних змінних. Додаються 6 додаткових ознак, наприклад, [monday, tuesday, wednesday, thursday, friday, saturday], які приймають значення 0 і 1 в залежності від дати. Ознака sunday, що відповідає сьомому дню тижня, не додають, тому що вона буде лінійно залежати від інших днів тижня і це буде впливати на модель.

Річна ж сезонність моделюється рядами Фур'є.

Тренд $g(t)$ - це кусково-лінійна або логістична функція. З лінійною функцією все зрозуміло. Логістична ж функція виду $g(t) = \frac{C}{1+exp(-k(t-b))}$ дозволяє моделювати зростання з насиченням, коли при збільшенні певного показника знижується темп його зростання. Типовий приклад - це зростання аудиторії програми або сайту.

Крім усього іншого, бібліотека вміє за історичними даними вибирати оптимальні точки зміни тренду. Але їх також можна задати і вручну (наприклад, якщо відомі дати релізів нової функціональності, які сильно вплинули на ключові показники).

Компонента $h(t)$ відповідає за задані користувачем аномальні дні, в тому числі і нерегулярні, такі як, наприклад, свята.

Помилка містить інформацію, яка не врахована моделлю.

## 2.6 Методи моделювання часових рядів на основі машинного навчання

Однією з фундаментальних проблем, яка тривалий час мучила традиційні архітектури нейронних мереж, була здатність інтерпретувати послідовності входів, які накладались один на одного для інформації та

контексту. Ця інформація може бути попередніми словами в реченні, щоб контекст передбачав, яким може бути наступне слово, або це може бути тимчасова інформація послідовності, яка дозволить контекст на елементах цієї послідовності, заснованих на часі.

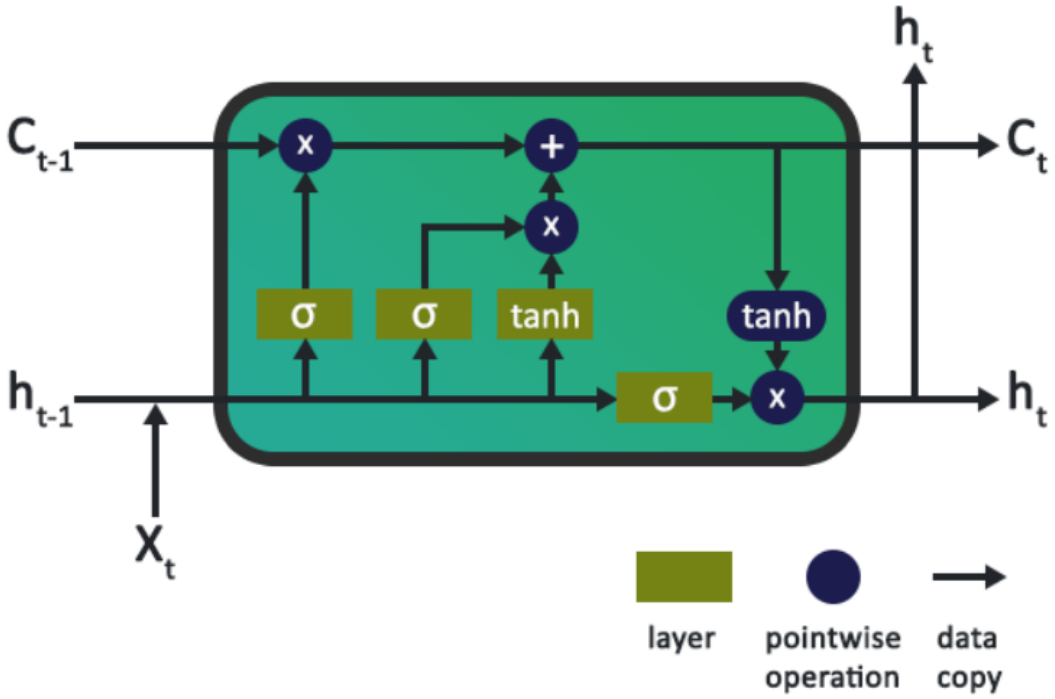
Простіше кажучи, традиційні нейронні мережі щоразу приймають автономний вектор даних і не мають поняття пам'яті, щоб допомогти їм у виконанні завдань, які потребують пам'яті.

Рання спроба вирішити цю проблему полягала у використанні простого підходу типу зворотного зв’язку для нейронів у мережі, де вихідний сигнал подавався назад на вхід, щоб надати контекст останнім побаченим входам. Вони називались періодичними нейронними мережами (RNN). Хоча ці RNN працювали, певною мірою, вони мали досить великий недолік: що будь-яке їх значне використання призводить до проблеми, яка називається проблемою зникаючого градієнта. Ми не будемо уточнювати проблему зникаючого градієнта, а лише зауважимо, що RNN погано підходять для більшості реальних проблем через цю проблему, отже, потрібно було знайти інший спосіб вирішення контекстної пам'яті.

Тут на допомогу прийшла нейронна мережа довгострокової пам’яті (LSTM). Як і нейрони RNN, нейрони LSTM зберігали контекст пам'яті у своєму конвеєрі, щоб дозволити вирішувати послідовні та тимчасові проблеми без проблеми зникаючого градієнта, що впливає на їх продуктивність.

В Інтернеті можна знайти багато дослідницьких статей та статей, які дуже детально обговорюють роботу нейронів LSTM. Однак у нашій роботі ми не будемо обговорювати складну роботу LSTM, оскільки нас більше турбує їх використання для вирішення наших проблем.

Для контексту, нижче наведена схема типової внутрішньої роботи нейрону LSTM. Він складається з декількох шарів і точкових операцій, які виконують роль входів для введення, виведення та забуття, які живлять стан комірки LSTM. Цей стан комірки є тим, що зберігає довгострокову пам’ять і контекст у мережі та на входах.




«Рис 2.3 Модель LSTM нейрона»

## 2.7 Порівняння ефективності прогнозування моделей

Щоб оцінити та порівняти точність прогнозування наших моделей прогнозування, потрібно добре перевірений набір спостережень в реальному часі. З цією метою був використаний відомий набір даних змагань М3. М3-змагання - це серія емпіричних досліджень, основна мета яких є полягає у порівнянні результативності різних методів часових рядів. Воно отримало значне розповсюдження з того часу, як було вперше організоване в 1982 році

групами, очолюваними дослідником Спіросом Макрідакісом. Набір даних для змагань охоплює широкий діапазон сфер, включаючи бізнес, економіку, макродані та промислові заходи. Ідея така: охоплюючи такий широкий діапазон полів, прогнози моделей можна точно оцінити незалежно від того, до якого домену застосовується модель. Для наших цілей ми в основному будемо працювати з економічною та діловою категоріями часових рядів набору даних.

Після публікації набору даних учасники конкурсу створили моделі та подавали свої найкращі прогнози, які були ними оцінені та які порівнюється за допомогою різноманітних методів.

Ці змагання проводились неодноразово, перший раз у 1982 році, другий у 1993 році та третій у 2000 (звідси і назва М3 для третього змагання). Четвертий конкурс був оголошений останнім - 2018 р. Кожного разу дослідники, які відповідальні за організацію заходу, ділились цікавими висновками, які вони отримали із свого досвіду. Результати теоретичних досліджень:

- ✔ Статистично складні або складні методи не обов'язково дають більш точні прогнози, ніж простіші.
- ✔ Відносний рейтинг результатів варіюється залежно від міри точності, що використовується.
- ✔ Точність поєднання різних методів перевершує в середньому показник методів використаних поодинці.
- ✔ Окремі методи поєднуються і дуже добре справляються з іншими методами. Точність різних методів залежить від тривалості прогнозування (Makridakis and Hibon, 2000).

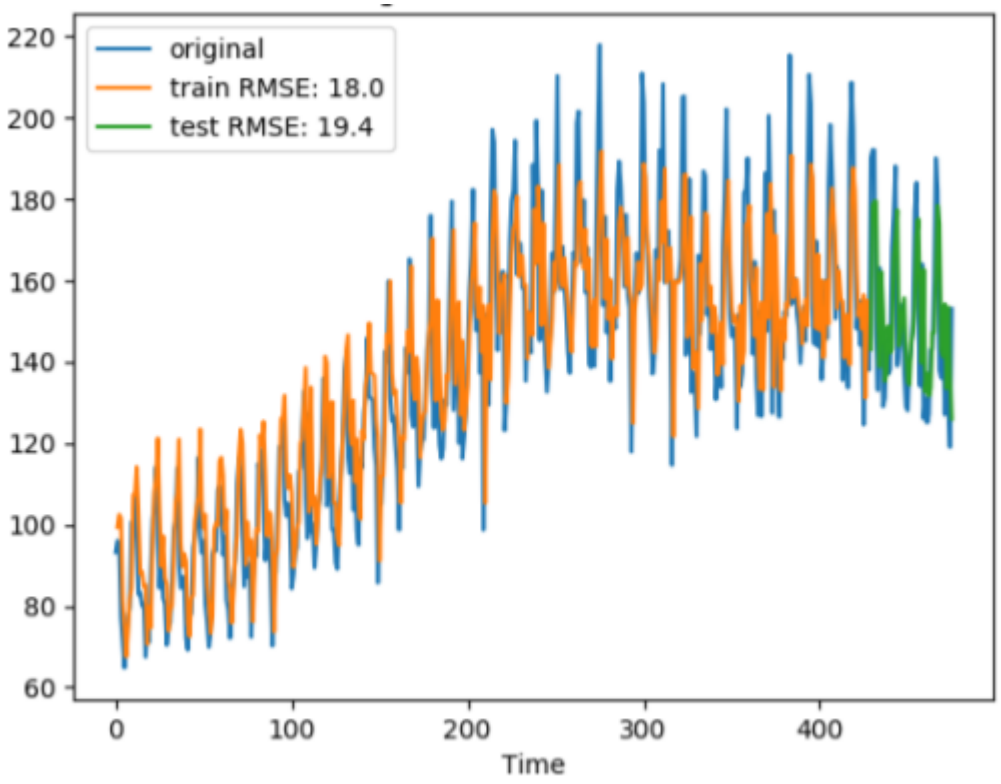
Щоб оцінити та порівняти точність прогнозування наших моделей прогнозування, нам потрібен добре перевірений набір спостережень у режимі реального часу. M3-змагання - це серія емпіричних досліджень, ціль яких полягає у порівнянні результативності різних методів часових рядів. Набір даних для змагань охоплює широкий діапазон сфер застосувань, включаючи бізнес, економіку, макродані та промислові заходи. Охоплюючи такий широкий діапазон даних, прогнози моделей можуть точно судити незалежно від того, до якого домену застосовується модель.

Для перевірки ефективності моделі застосуємо такі метрики, як SMAPE - симетричний середній відсоток відхилення, MdRAE - середня відносна абсолютна помилка та RMSE – середньо квадратичне відхилення.

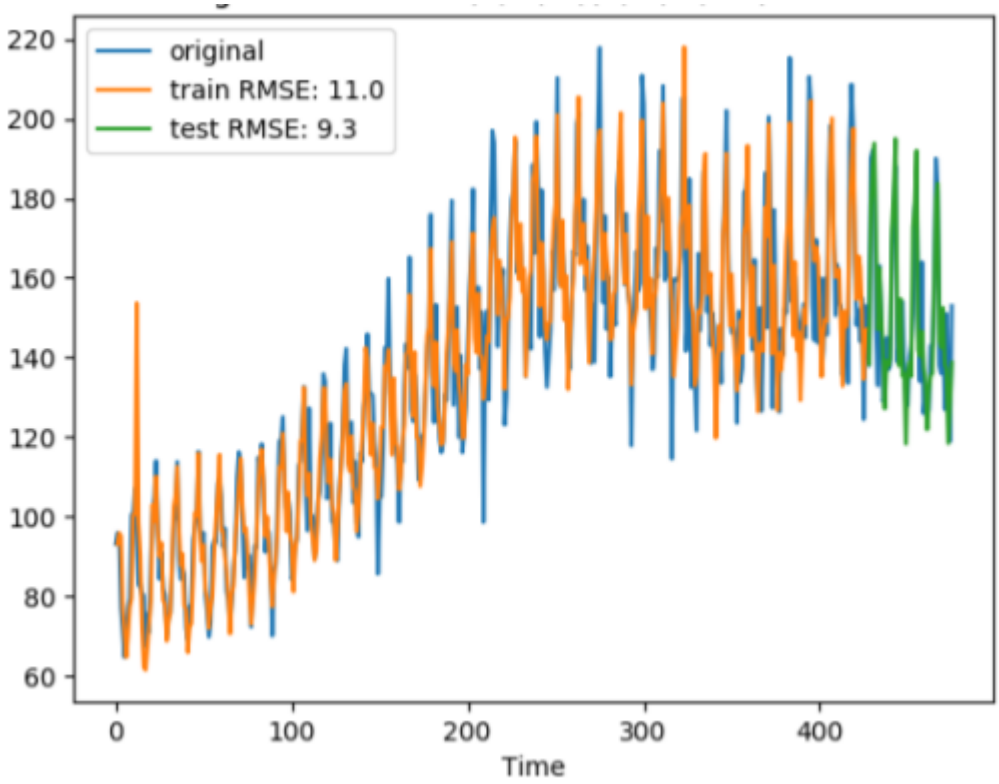
Результатами проходження тестів стали:

| | MdRAE | RMSE | SMAPE |
|---|---|---|---|
| ARIMA Train | 11.706 | 11.005 | 5.998 |
| ARIMA Test | 9.641 | 9.252 | 5.054 |
| LSTM Train | 14.565 | 17.995 | 10.271 |
| LSTM Test | 14.988 | 19.363 | 10.416 |

«Рис 2.4 Метрики похибок для моделювання попиту на пиво у Австрії»




«Рис 2.5 Графік порівняння моделі і реальних даних для LSTM»

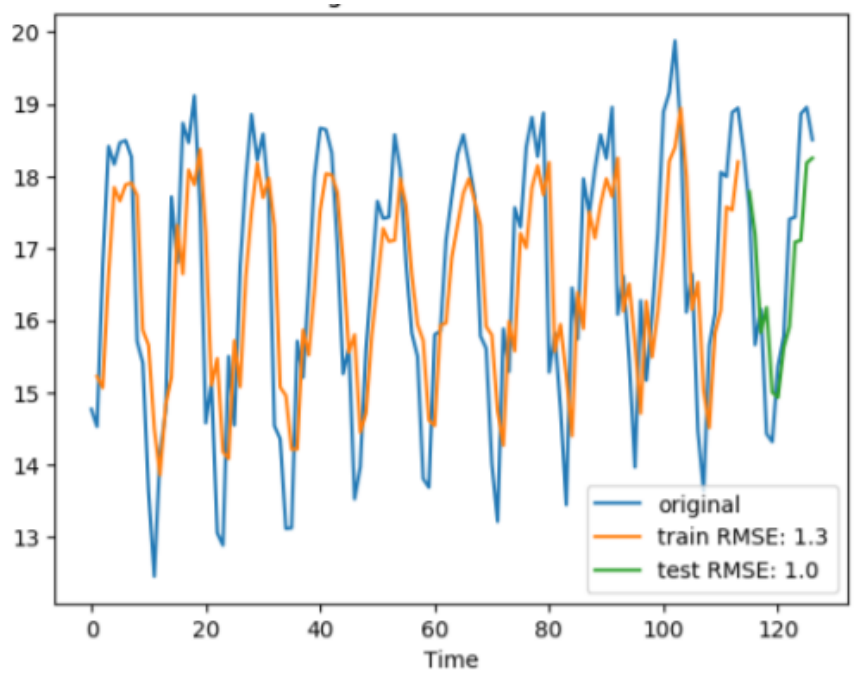



«

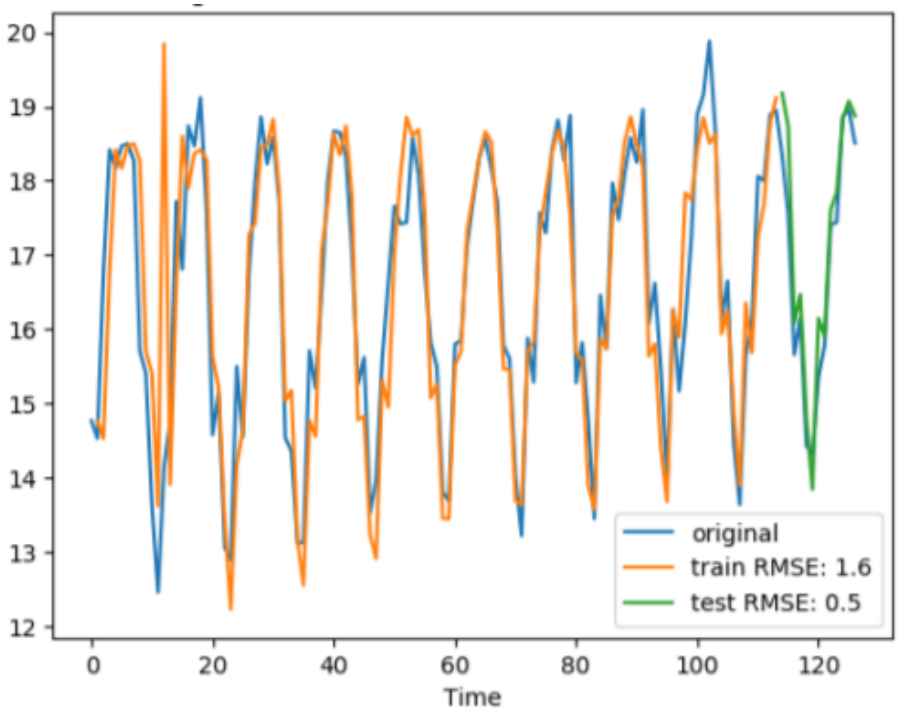
Рис 2.6 Графік порівняння моделі і реальних даних для ARIMA»

| | MdRAE | RMSE | SMAPE |
|---|---|---|---|
| ARIMA Train | 0.75 | 1.65 | 5.347 |
| ARIMA Test | 0.797 | 0.535 | 2.571 |
| LSTM Train | 1.518 | 1.29 | 6.714 |
| LSTM Test | 1.527 | 1.014 | 4.907 |

«Рис 2.8 Метрики похибок для моделювання попиту на пиво у США»




«Рис 2.7 Графік порівняння моделі і реальних даних для LSTM»




«Рис 2.7 Графік порівняння моделі і реальних даних для ARIMA»

Була розглянута ефективність двох методів для прогнозування часових рядів. ARIMA є одним із багатьох традиційних методів, які покладаються на основну стохастичну модель для вилучення інформація з даних. LSTM представляють алгоритмічний підхід до аналізу часових рядів даних і трактує базовий процес як невідомий. Для порівняння ми використовували дані з різних джерел, включаючи дані про попит на пиво з США та Австралії та анонімізований час серія з конкурсу M3. Наш основний критерій оцінки - це те, наскільки добре працює модель на даних поза вибіркою, виміряних різними показниками похибки. Ми дійшли висновку, що для даних з сильною сезонністю, обидві моделі прогнозують досить ефективно, а модель ARIMA видає кращий результат, але більш вибаглива до початкових даних.

Якщо порівнювати дані результат із моделлю Prophet, то Prophet має значно гірші результати відносно інших через те, що набір даних є недостатнім для даної моделі, через її регресійну природу, для якої потрібен великий набір даних, щоб отримати ефективний результат.

# РОЗДІЛ 3 Апаратно-програмна реалізація системи

## 3.1 Архітектура та елементи системи

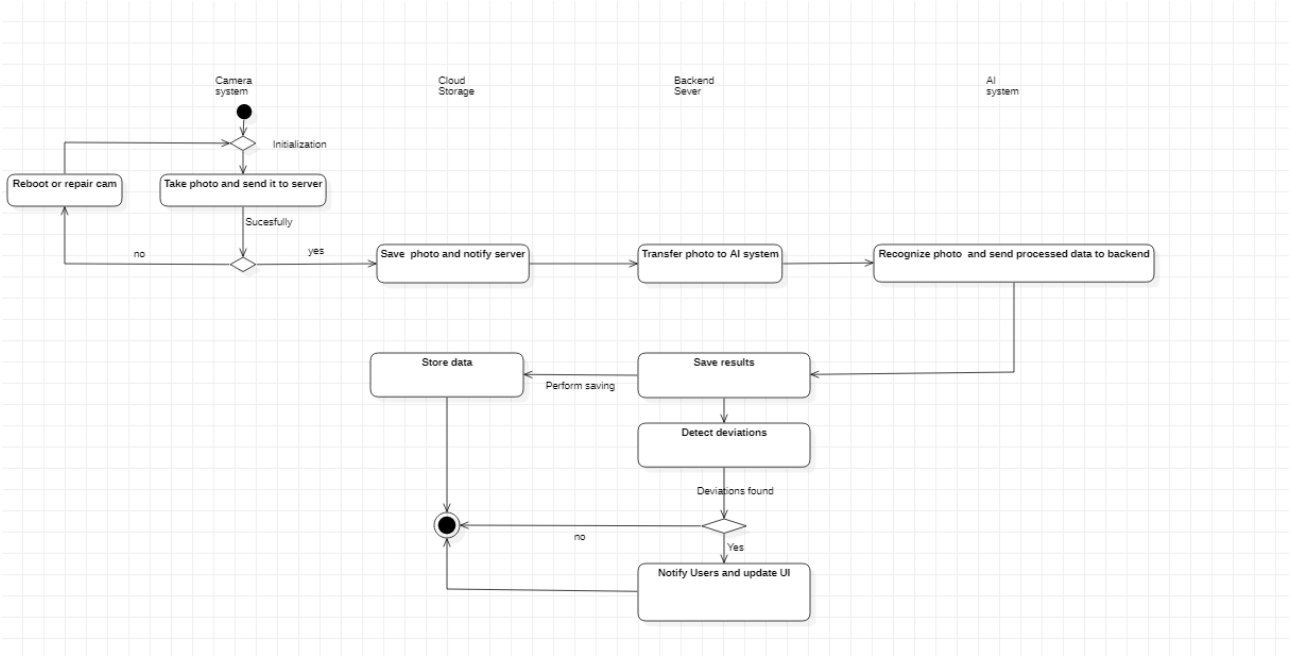
Основна концепція продукту:

- Система камери відстежує зміни на полиці та надсилає фотографії в хмарне сховище
- Хмарне сховище
- Нейронні мережі, що аналізують фотографії та отримують з них дані
- Бекенд-веб-служба обробляє інформацію, отриману від нейронних мереж, і повідомляє персонал замовника у разі невідповідності показників на полицях
- Веб-інтерфейс, який відображає сповіщення та статистику
- Боти месенджерів, які доставляють інформацію персоналу
- Система прогнозування попиту на товари на основі обороту

Система камер робить фотографії за попередньо визначений проміжок часу та надсилає їх у хмарне сховище. Наприклад, можна робити фотографії кожні 10 хвилин для моніторингу ООС або пару разів на день для перевірки планограми та ціни.

Система створена для роздрібної торгівлі та логістичних компаній, які не мають сучасних інструментів для контролю стану полиці. Вони не орієнтовані на технології, тому їм важко впроваджувати інновації у свої бізнес-процеси. Отже, було створено гнучке рішення, яке можна легко інтегрувати та яке,допомагає компаніям продавати більше. Рішення

забезпечує моніторинг полиць у режимі реального часу та надає таку інформацію, як OSA, невідповідність планограм, застарілі ціни. У випадку OOS система повідомляє персонал автоматично, тому їм не потрібно регулярно перевіряти полиці вручну. Це допомагає мінімізувати цю метрику та заощадити час співробітників. З іншого боку продукт - це повністю автоматизована система без людського фактора, тому вона позбавлена фактору потенційного шахрайства.




«Рис 3.1 UML діаграма активності, що показує логіку взаємодії сервісів»

## 3.2 Реалізація системи камер

### 3.2.1 Створення технічних вимог для системи камер

Одним із основних джерел інформації для сервісу є камери спостереження за полицями, оскільки система отримує інформацію про таку метрику, як процент заповнюваності полиці із камер. Результат постійного моніторингу товару є така метрика, як оборот товару, що на наступних кроках використовуються для прогнозування попиту на певний товар.

До системи камер було висунуто певний набір вимог:

- Камера повинна мати достатню роздільну здатність для розпізнавання товарів і цін на товари
- Камера повинна мати низьку ціну
- Камера повинна самостійно відправляти фотографії через мережу інтернет
- Камера повинна мати доступ до мережі інтернет

В результаті експериментів, вивчення документації та функціоналу мікроконтролерів із підтримкою моделей камер був вибраний ESP-32, через наявність на платі Wifi модуля, що вирішує проблему доступу до мережі інтернет без використання LAN кабелів і також вбудований 2 мега піксельний сенсор OV2640.


«Рис 3.1 ESP32-cam module»

### 3.2.2 Платформа ESP-32

Розроблена Espressif Systems, ESP32 - це недорога система з низьким енергоспоживанням на мікросхемі (SoC), що має Wi-Fi та подвійним режимом Bluetooth. Сімейство ESP32 включає мікросхеми ESP32-D0WDQ6 (і ESP32-D0WD), ESP32-D2WD, ESP32-S0WD та систему в пакеті (SiP) ESP32-PICO-D4. В його основі - двоядерний або одноядерний мікропроцесор Tensilica Xtensa LX6 з тактовою частотою до 240 МГц. ESP32 високо інтегрований із вбудованими антенними вимикачами, підсилювачем потужності, підсилювачем прийому з низьким рівнем шуму, фільтрами та модулями управління потужністю. Розроблений для мобільних пристроїв, переносної електроніки та додатків IoT, ESP32 досягає наднизького

енергоспоживання завдяки функціям енергозбереження, включаючи точне регулювання годинника з чіткою роздільною здатністю, множину режимів живлення та динамічне масштабування навантаження.

### 3.2.3 Реалізація функціоналу системи камер на основі плати ESP32-cam

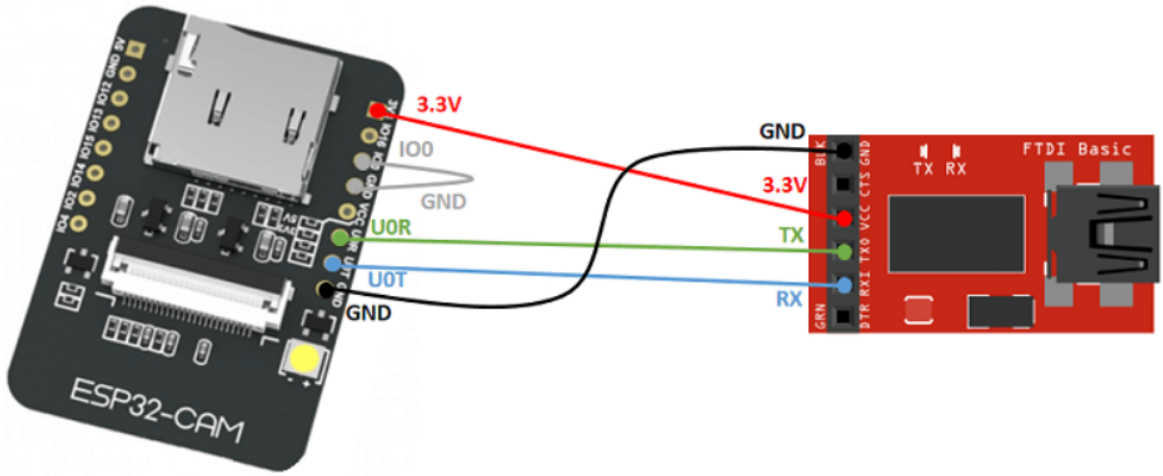
Для того, щоб задовільнити вимоги до системи камер потрібно, щоб камера виконувала наступні кроки:

1) Ініціалізація
2) Підключення до мережі Wifi
3) Отримання фотографії із модуля камери
4) Відправка фотографії на сервер
5) Перехід у режим сну

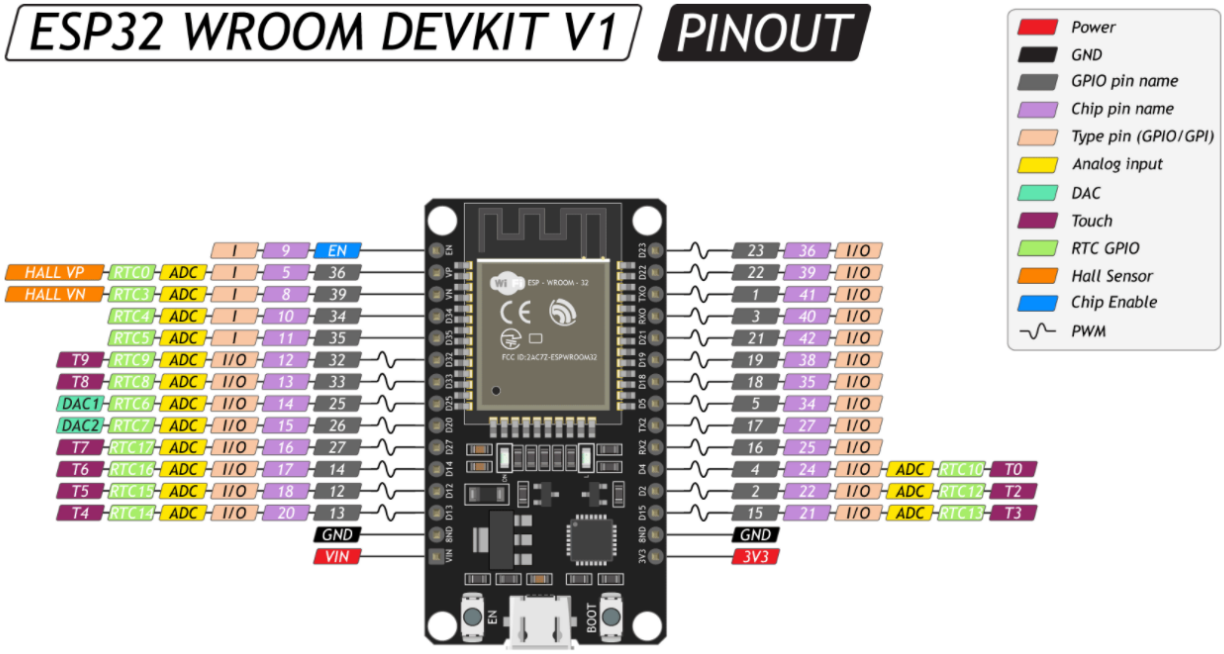
Для створення програм для даного чіпу компанія виробник створила фреймворк під назвою ESP-IDF. Даний фреймворк дозволяє записувати програмний код та взаємодіяти з чіпом через серіал порти комп'ютера. Для створення коду було використане інтегроване середовище для розробки Arduiono IDE та мова c++.

Для того, щоб записати код на чіп потрібний спеціальний пристрій під назвою програматор, оскільки платформа ESP-32 є мікроконтролером. Для запису можна використати програматор: USB/UART converter FTDI FT232, miniUSB.

Для запису коду на чіп потрібно підключити його наступним чином:




«Рис 3.2 Налаштування підключення програматора до ESP-32»




«3.3 Структура контрактів на платі ESP-32»

С двох сторін плати розташовані контактні гребінки по 15 пінів з кроком 2,54 мм (модифікація на 30 пінів).

Доступні 25 контактів загального призначення. Всі контакти підтримують переривання. Максимальний струм на контактах: 12 мА.

Цифрові 21 контакт вводу-виводу (GPIO): 1-5, 12-19, 21-23, 25-27, 32 і 33. Контакти загального призначення. Піни можуть бути налаштовані на вхід або на вихід. Логічний рівень одиниці - 3,3 В, нуля - 0 В. Максимальний струм виходу - 12 мА. Всі висновки введення-виведення можуть працювати як ШІМ, що дозволяє виводити аналогові значення в вигляді ШІМ-сигналу з розрядність 16 біт. Максимальна кількість каналів 16

цифрові 4 контакти введення (GPI): 34, 35, 36 і 39. Можуть бути налаштовані тільки на вхід.

15 аналогових входів з АЦП (12 біт): 2, 4, 12-15, 25-27, 32-36 і 39. Дозволяє уявити аналогову напругу в цифровому вигляді з розрядністю 12 біт.

2 аналогових виходу з ЦАП (8 біт): 25 (DAC1) і 26 (DAC2). Аналоговий вихід цифро-аналогового перетворювача, який дозволяє формувати 8-бітові рівні напруги. Висновки можуть використовуватися для аудіо-виходу.

10 контактів ємнісного сенсора

На контактах вводу-виводу можна конфігурувати апаратні інтерфейси:

3 × UART

3 × SPI

2 × I²C

3 × I²S

Код програмного забезпечення для камери:

```
#include "Arduino.h"

// Enable Debug interface and serial prints over UART1
#define DEGUB_ESP 1
#define uS_TO_S_FACTOR 1000000

#include <WiFi.h>
#include <WiFiClient.h>
#include <NTPClient.h>
#include <WiFiUdp.h>

#include "esp_camera.h"
#include "esp_timer.h"
#include "img_converters.h"
#include "fb_gfx.h"

#include "fd_forward.h"
#include "fr_forward.h"
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "ESP32_FTPClient.h"

#include "config.h"


#ifdef DEGUB_ESP
#define DBG(x) Serial.println(x)
#else
#define DBG(...)
#endif

camera_fb_t *fb = NULL;
String pic_name = "";
String mac = WiFi.macAddress();
```

```
RTC_DATA_ATTR uint64_t bootCount = 0;

WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP);

uint64_t start_processing_time;

void deep_sleep(void);
void FTP_upload(void);
bool take_picture(void);
boolean camera_init(void);
boolean connect_wifi(void);

void setup()
{
  pinMode(4, OUTPUT);
  digitalWrite(4, LOW);
  for (int i = 0; i < FTP_COUNT; i++) {
    DBG(ftps[i].user);
  }
#ifdef DEGUB_ESP
  Serial.begin(115200);
  Serial.setDebugOutput(true);
#endif
}

void loop()
{
  // Take picture
  start_processing_time = millis();
  if (camera_init() && connect_wifi() && take_picture() )
  {
    FTP_upload();
  }
```

```
  deep_sleep();

}

boolean camera_init() {
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); //disable brownout detector
  mac.replace(":", "");
  mac.toLowerCase();
  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;
  //init with high specs to pre-allocate larger buffers
  if(psramFound()){
    config.frame_size = FRAMESIZE_QXGA;
    config.jpeg_quality = 10;
```

```
  } else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 10;
    config.fb_count = 1;
  }

  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK)
  {
    Serial.print("Camera init failed with error 0x%x");
    DBG(err);
    return false;
  }

  // Change extra settings if required
//  sensor_t * s = esp_camera_sensor_get();
  //s->set_vflip(s, 0);       //flip it back
  //s->set_brightness(s, 1);  //up the blightness just a bit
//  s->set_saturation(s, +1); //lower the saturation
  return true;
  }

void deep_sleep()
{
  esp_sleep_enable_timer_wakeup( SECONDS_TO_SLEEP * uS_TO_S_FACTOR - (millis() - start_processing_time) );
  DBG("Going to sleep after: " + String( millis() ) + "ms");
  WiFi.disconnect();
  Serial.flush();
  esp_camera_deinit();

  esp_deep_sleep_start();
  ESP.restart();
}



bool take_picture()
{
  unsigned long getting_time_start = millis();
  while (!timeClient.update() && millis() - getting_time_start < 30000) {
    timeClient.forceUpdate();
  }
  DBG("Taking picture now");
  fb = esp_camera_fb_get();
  if (!fb)
  {
    DBG("Camera capture failed");
    return false;
  }

  // Rename the picture with the time string
  pic_name = "photo-" + mac + "_" + String(timeClient.getEpochTime()) + ".jpg";
  DBG("Camera capture success, saved as:");
  DBG( pic_name );

  return true;
}

void FTP_upload()
{
  DBG("Uploading via FTP");

  for (int i = 0; i < FTP_COUNT; i++) {
    ESP32_FTPClient ftp (const_cast<char*>(ftps[i].server.c_str()), const_cast<char*>(ftps[i].user.c_str()), const_cast<char*>(ftps[i].password.c_str()));
    ftp.OpenConnection();
    ftp.InitFile("Type I");
    const char *f_name = pic_name.c_str();
    ftp.NewFile( f_name );
    DBG(fb->len);
    ftp.WriteData(fb->buf, fb->len + 1024);
```

```
    ftp.CloseFile();
    ftp.CloseConnection();
  }
  pic_name = "";

}

boolean connect_wifi() {

  int networks_count = WiFi.scanNetworks();
  for (int i = 0; i < networks_count; i++) {
    if (WiFi.isConnected()) break;
    for (int j = 0; j < WIFI_COUNT; j++) {
      if (WiFi.isConnected()) break;
      if (wifies[j].ssid == WiFi.SSID(i)) {
        DBG("\n" + wifies[j].ssid + " - FOUND");
        uint64_t connection_start = millis();
        WiFi.begin( wifies[j].ssid.c_str(), wifies[j].pass.c_str() );
        DBG("\nConnecting to WiFi");

        while ( WiFi.status() != WL_CONNECTED && millis() - connection_start < CON_TIMEOUT )
        {
          delay(500);
          DBG(".");
        }
      }
    }
  }

  return WiFi.isConnected();
}
```

## 3.3 Реалізація системи прогнозування часових рядів

### 3.3.1 Створення технічних вимог для системи прогнозування часових рядів

Після певного часу спостережень за полицею можна отримати інформацію про оборот товарів на ній. Для того, щоб оптимізувати логістичні процеси користувача, потрібно мати можливість гнучкого прогнозування попиту на товар на основі методів моделювання часових рядів. Основні вимоги до системи прогнозування в умовах задачі оптимізації бізнес процесів рітейлу:

1. Мати можливість масштабування при збільшення кількості продуктів.
2. Використовувати універсальний підхід, який можна використовувати незалежно від типу товару.

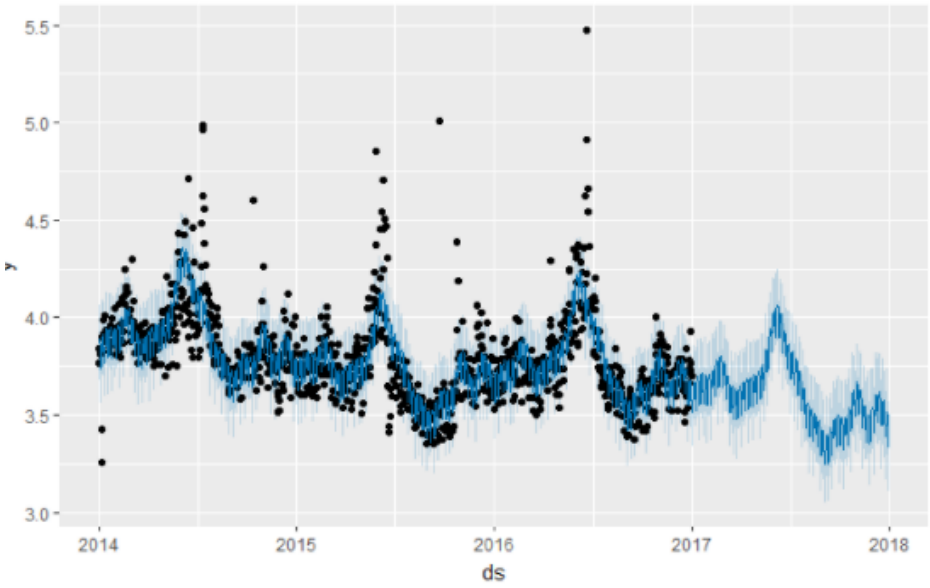
3. Враховувати сезонні показники.
4. Можливість динамічно оновлювати результати із отриманням нових даних.

Аналізуючи другу вимогу можна відкласти класичні підходи, оскільки ми не можемо сподіватися на те, що дані, які ми отримуємо будуть відповідати ознакам монотонності, а мануальні дії для зведення до монотонності і згладжування не масштабуються на 1000 товарів.

Оскільки і LSTM – нейронні мережі, і Prophet можуть бути достатньо гнучкими для забезпечення другого пункту, то вони задовольняють цей пункт і перший також.

Недоліком LSTM нейронных мереж е те, що вони показують високий результат на основі невеликої кількості спостережень, однак при значних об’ємах спостережень можуть втрачати ефективність через свій принцип навчання. Отже, у результаті оптимальним варіантом при довгостроковому наборі даних буде підхід Prophet.

Також використання LSTM нейронних мереж є дуже ресурсоємною задачею, особливо навчання на великих масивах даних.




«Рис 3.4 Автоматичний прогноз, використовуючи Prophet»

### 3.3.2 Організація системи з точки зору програмної архітектури

Можливість безболісного масштабування є однією із основних цілей створення даної системи. Оскільки різні задачі системи можуть містити зовсім різні з точки зору навантаження задачі усі сервіси реалізовані використовуючи мікросервісну архітектуру, тобто кожен із модулів не має залежностей від інших, а взаємодія між сервісами забезпечена за допомогою окремого сервісу, що відповідає за розподіл навантаження і задач між усіма іншими, використовуючи принцип черги подій.

# ВИСНОВКИ

В ході виконання кваліфікаційної роботи на здобуття ступеня магістра мною було:

1. Проведено аналіз та порівняння ефективності статистичних моделей та моделей на основі нейронних мереж.
2. Проведено адаптацію та інтеграцію інструменту для моделювання часових рядів для вирішення прикладної задачі моделювання обороту товарів.
3. Створено систему для отримання візуальної інформації для подальшої обробки із використання мікроконтролерів.
4. Отримано практичні навички із розробки та моделювання програмного забезпечення.
5. Поглиблено теоретичні та прикладні знання у таких напрямах, як штучний інтелект та аналіз даних.

## Література:


1. Практикум по эконометриці И.И. Елисеева - М.: Фінанси і статистика, 2003. - 192 с.
2. Мішуліна О. А. Статистичний аналіз і обробка часових рядів. - М .: МІФІ, 2004. - С. 180. - ISBN 5-7262-0536-7.
3. Бокс Дж., Дженкінс Г. Аналіз часових рядів прогноз і управлення.
4. Mcdowall, Mccleary, Meidinger and Hay (1980) Аналіз перериваємих часових рядів
5. Веллеман і Хоглін Програми, основи та обчислення дослідницького аналізу даних
6. Sean J. Taylor, Benjamin Letham "Прогнозування при масштабуванні
7. Етем Альпайдін. Вступ до машинного навчання, друге видання. Адаптивне обчислення та машинне навчання. MIT Press, лютий 2010 р.
8. Ульріх Андерс та Олаф Корн. Вибір моделі в нейронних мережах. Нейронна мережа.
9. Роберт Р. Андравіс, Амір Ф. Атія та Хішам Ель-Шишині. Поєднання довгострокових та короткострокових прогнозів із застосуванням для прогнозування туристичного попиту. Міжнародний журнал прогнозування, у пресі, виправлений доказ: -, 2010
10. Роберт Р. Андравіс, Амір Ф. Атія та Хішам Ель-Шишині. Прогнозні комбінації обчислювального інтелекту та лінійних моделей для змагань з прогнозування часових рядів nn5. Міжнародний журнал прогнозування, у пресі, виправлений доказ: -, 2011
11. Амір Атія, Сюзан М. Ель-шура, Самір І. Шахін та Мохамед С. Ель-шериф. Порівняння між методами прогнозування нейронних мереж - тематичне

дослідження: Прогнозування річкових потоків. Транзакції IEEE щодо нейронних мереж, 1999.

12. Ч.Г. Аткесон, А. В. Мур та С. Шаал. Локально зважене навчання. Огляд штучного інтелекту
13. Пол Дж. Вербос. Узагальнення зворотного поширення із застосуванням до періодичної моделі ринку газу. Нейронні мережі 1988
14. Йорг Д. Вічард. Прогнозування часових рядів nn5 за допомогою гібридних моделей. Міжнародний журнал прогнозування, у пресі, виправлений доказ: 2010
15. Г. Пітер Чжан та Мін Ці. Прогнозування нейронної мережі для сезонних та тенденцій часових рядів. Європейський журнал операційних досліджень,
16. Paul J. Werbos. Generalization of backpropagation with application to a recurrent gas market model. Neural Networks, 1(4):339 – 356, 1988. ISSN 0893-6080. doi: DOI:10.1016/0893-6080(88)90007-X. URL http://www.sciencedirect.com/science/article/B6T08-485RHDS-7/2/037e956cda49bd2d2c66085cfccd7de4.
17. A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, and A. Lendasse. Methodology for long-term prediction of time series. Neurocomputing, 70(16-18):2861–2869, October 2007. doi: 10.1016/j.neucom.2006.06.015.
18. S.B. Taieb, A. Sorjamaa, and G. Bontempi. Multiple-output modeling for multi-step-ahead time series forecasting. Neurocomputing, 73(10-12):1950 – 1957, 2010. ISSN 0925-2312. doi: DOI: 10.1016/ j.neucom. 2009.11.030. URL http://www.sciencedirect.com/science/article/B6V10-4YJ6GCW-4/2/8429b80db7773717c9d455b485fb7c4d. Subspace Learning / Selected papers from the European Symposium on Time Series Prediction.