

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЇВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем факультету інформатики

ЗАСТОСУВАННЯ НЕЙРОМЕРЕЖЕВИХ АЛГОРИТМІВ ДЛЯ ПРОГНОЗУ ЧАСОВИХ РЯДІВ

**Текстова частина до курсової роботи
за спеціальністю „Комп’ютерні науки” 122**

Керівник курсової
роботи

к.т.н., доц.
Олецький О.В.

(підпис)

“ ”

2024 р.

Виконала студентка
Томенко Н.Д

“ ”

2024 р.

Київ 2024

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЇВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ
Зав. кафедри мультимедійних
систем,
проф., д.ф.-м.н.
_____ О. П. Жежерун
(підпис)
„_____” _____ 2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу

студентці Томенко Наталі факультету інформатики 3-го курсу
ТЕМА Застосування нейромережових алгоритмів для прогнозу часових
рядів

Зміст ТЧ до курсової роботи:

- Індивідуальне завдання
- Вступ
- Розділ 1. Теоретичні основи часових рядів та їх прогнозування
- Розділ 2. Основні теоретичні відомості про нейронні мережі
- Розділ 3. Прогнозування даних часового ряду за допомогою
нейромережових моделей
- Висновки
- Список використаної літератури

Дата видачі «___» _____ 2023 р.

Керівник _____
(підпис)

Завдання отримав _____
(підпис)

Тема: Застосування нейромережових алгоритмів для прогнозу часових рядів

Календарний план виконання роботи:

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	15.10.2023	
2.	Пошук та огляд літератури за темою роботи.	Жовтень – грудень 2023	
3.	Дослідження сучасних методів та підходів до вирішення завдання	Січень - лютий 2024	
3.	Пошук набору даних для задачі прогнозування	Березень 2024	
4.	Застосування нейромережових алгоритмів для прогнозування часового ряду із обраного набору даних	Квітень 2024	
5.	Написання текстової частини роботи	01.05.2024	
6.	Перегляд курсової роботи науковим керівником	05.05.2024	
7.	Внесення змін до курсової роботи відповідно зауважень наукового керівника.	10.05.2024	
8.	Створення презентації до курсової роботи	13.05.2024	
9.	Захист курсової роботи	22.05.2024	

Студент: Томенко Наталя Дмитрівна

Керівник: Олецький Олексій Віталійович

“ _____ ”

Зміст

Вступ	5
Розділ 1. Теоретичні основи часових рядів та їх прогнозування	7
1.1 Поточний стан питання	7
1.2 Основні відомості про часові ряди	7
1.3 Процес прогнозування часових рядів	8
1.3.1 Підготовка та аналіз часових рядів	8
1.3.2 Вибір та підгонка моделі	10
1.3.3. Перевірка моделі	11
1.3.4. Застосування та моніторинг роботи моделі	12
1.4 Моделі для прогнозування часових рядів	12
Розділ 2. Основні теоретичні відомості про нейронні мережі	14
2.1 Основні відомості про нейронні мережі	14
2.2. Рекурентні нейронні мережі (RNN)	15
2.3 Мережі довготривалої короткочасної пам'яті (LSTM)	17
2.4 Вентильний рекурентний вузол (GRU)	19
Розділ 3. Прогнозування даних часового ряду за допомогою нейромережових моделей	21
3.1 Датасет	21
3.2 Обрані технології	22
3.3 Підготовка та дослідження даних часового ряду	22
3.3 Створення та тренування моделей	25
3.4 Оцінка моделей	25
Висновки	28
Список використаної літератури	29

Вступ

У сучасному світі, де дані та їх аналіз є основою прийняття рішень, можливість передбачати майбутні події, закономірності та тренди стає все більш цінною. В тому числі це включає прогнозування даних часових рядів. В повсякденному житті ми досить часто стикаємось з часовими рядами: прогноз погоди, коливання курсу валют, електрокардіограма, тощо. Сфер застосування часових рядів безліч, і задача їх прогнозування є актуальною та важливою для прийняття зважених та інформованих рішень.

Для прогнозування часових рядів в різні часи була запропонована велика кількість різноманітних підходів і моделей, включаючи використання нейронних мереж. Зокрема, особливо ефективними виявилися рекурентні нейронні мережі (RNN) та їхні різновиди завдяки своїй можливості використання результатів попередніх кроків для генерації наступних, що чудово підходить для послідовної природи даних. Хоча нейронні мережі наразі й не здатні замінити усіх інших існуючих методів прогнозування часових рядів, вони посідають заслужене місце в інструментарії вирішення цієї задачі.

Мета і завдання дослідження:

Метою даної роботи є дослідження прогнозування часових рядів, рекурентних нейронних мереж та їх використання в таких типах завдань, а також перевірка прогнозування нейромережевих алгоритмів на реальних даних.

Завдання дослідження:

1. Розглянути теоретичні основи часових рядів та їх прогнозування
2. Розглянути різновиди нейронних мереж що використовуються для прогнозування часових рядів

3. Підібрати набір даних для подальшого прогнозування
4. Провести прогнозування обраних даних для різних нейромережових алгоритмів
5. Провести аналіз отриманих результатів

Розділ 1. Теоретичні основи часових рядів та їх прогнозування

1.1 Поточний стан питання

Історично прогнозування часових рядів покладалося на лінійні моделі, такі як авторегресійне інтегроване ковзне середнє (ARIMA) і експоненціальне згладжування (ES). Ці методи, ефективні для простих наборів даних, домінували до тих пір, поки перевагу не взяли передові методи машинного навчання, які забезпечили більшу гнучкість та могли обробляти складні нелінійні моделі.

Трансформація поглибилася з появою методів глибинного навчання, зокрема рекурентних нейронних мереж (RNN) та їх різновидів. Нейронні мережі чудово моделюють часові ряди завдяки своїй здатності виявляти довгострокові залежності, що має велике значення для точного прогнозування.

Загалом наразі неможливо виділити якийсь один підхід який би домінував та випереджав усі інші. Часто можна зустріти також і поєднання надійних класичних методів із застосуванням сучасних нейронних мереж. Сфера продовжує розвиватися, а вибір щодо використання певного підходу робиться на основі поставленої задачі, бажаної точності прогнозування та особливостей вхідних даних.

1.2 Основні відомості про часові ряди

Часовий ряд – це послідовність точок даних, записаних протягом певного часу, як правило, через однакові інтервали. Ці точки даних можуть представляти будь-яку змінну, яка змінюється з часом, наприклад, ціни на акції, показники температури або споживання електроенергії. Основною характеристикою даних часових рядів є їх часовий порядок, який відрізняє їх від інших типів даних важливістю часу спостережень.

Будь-який часовий характеризується чотирма компонентами:

1. Тренд: плавна довгострокова тенденція. Тренд відображає загальну схильність даних до збільшення або зменшення протягом тривалого періоду часу.
2. Сезонність: визначає закономірність, що повторюється протягом певного періоду. Аби називатись сезонністю закономірність має повторюватись періодично.
3. Циклічність: також є закономірністю, що повторюється протягом певного періоду, проте, на відміну від сезонності, повторюється аперіодично, і охоплює більш тривалі проміжки часу.
4. Шум: шумом називають абсолютно випадкове коливання даних, що неможливо передбачити.

Якщо ряд не має в собі закономірностей, то він є стаціонарним. Окремо також виділяють часові ряди, що не можна перетворити на стаціонарні шляхом позбавлення їх сезонності та/або тренду. Такі ряди називають різницево-стаціонарними (difference – stationary). [1]

Аби дослідити тест на стаціонарність використовують тест Дікі-Фуллера, що працює за методом знаходження одиничного кореня (unit root). Одиничний корінь — це властивість часового ряду, яка вказує на наявність у часовому ряді стохастичної тенденції, що відхиляє його від середнього значення. Наявність одиничного кореня робить часовий ряд нестаціонарним і, як наслідок, це призводить до труднощів у отриманні статистичних висновків із часового ряду та майбутніх прогнозів. [2]

1.3 Процес прогнозування часових рядів

1.3.1 Підготовка та аналіз часових рядів

Аналіз часових рядів має фундаментальну роль для їхнього подальшого прогнозування, оскільки він надає методи, необхідні для

розуміння, моделювання та прогнозування поведінки залежних від часу даних. Він допомагає визначити основні компоненти часового ряду, а саме тренд, сезонність і циклічність. Їх виокремлення є необхідним для вибору та побудови оптимальної моделі прогнозування.

Аналізу часових рядів зазвичай передують підготовка та попередня обробка даних. Конкретний алгоритм підготовки залежить від самого вхідного набору даних. Це може включати базову чистку даних і нормалізацію, заповнення пропущених значень, а також приведення даних часового ряду до певного вигляду.

1. *Описовий аналіз*: на початкових етапах аналітики часто використовують лінійні графіки, гістограми та підсумкову статистику для візуалізації даних. Це потрібно для попереднього ознайомлення з даними та дослідження їхніх закономірностей. На цьому етапі можна визначити розподіл часового ряду, наявність викидів, та ідентифікувати основні характеристики часового ряду про які йшлося раніше.

2. *Декомпозиція*: декомпозиція — це метод розбиття часового ряду на окремі компоненти — тренд, сезонність і шум. Виділяючи ці компоненти, значно ідентифікувати закономірності в даних, а отже і розробити інструментарій для подальшого прогнозування. [3]

3. *Моделювання змінних (feature engineering)*: для застосування методів машинного навчання для прогнозування часових рядів, їхні дані мають бути перетворені на набір даних для контрольованого навчання. За замовчуванням, часовий ряд не має залежних і незалежних змінних. Потрібно ж сконструювати набір даних так, щоб він мав цільову змінну, яку потрібно передбачити, та вхідні змінні за якими це передбачення буде відбуватися. Один з ключових підходів до трансформації часових рядів це застосування лагів (feature lags). За цього підходу значення змінної в часі (t)

використовується для прогнозування значення в наступній точці часу ($t+1$). [3]

4. Дослідження автокореляції: кореляцією називають зв'язок змінних між собою. У часових рядах це також називається автокореляцією, оскільки ми обчислюємо кореляцію проти значень лагу того самого ряду. Побудова графіка автокорельованих даних дає графік, подібний до синусоїдальної функції. [4] Якщо автокореляція присутня, то можна використовувати поточні значення змінної для кращого передбачення майбутніх.

Важливо зауважити, що жоден із цих кроків не є обов'язковим при аналізі часового ряду, порядок та конкретні складові аналізу визначаються залежно від потреб, обраної моделі та особливостей вхідних даних.

1.3.2 Вибір та підгонка моделі

Вибір і підгонка моделі складається з вибору однієї або кількох моделей прогнозування та підгонки моделі до даних часового ряду. Моделювання часових рядів має на меті представити дані таким чином, щоб відобразити їх структуру з якомога меншою похибкою. Під підгонкою мається на увазі підбір значень її невідомих параметрів. Метою цього підбору є мінімізація похибки моделі, тобто різниці між фактичними значеннями та значеннями, передбаченими моделлю. Зазвичай це робиться шляхом оптимізації певної цільової функції, наприклад мінімізації суми квадратів помилок (SSE) або максимізації функції ймовірності.

Після того, як модель підігнана до навчальних даних, її можна буде використовувати для прогнозування нових, невідомих даних. Важливо зазначити, що при підгонці до існуючих даних потрібно бути обережними аби уникнути ситуації *overfitting* чи *underfitting*. *Overfitting* називають випадок коли модель замість загальних закономірностей фіксує шум у

навчальних даних, а *underfitting* – коли, навпаки, модель є занадто простою і недостатньо фіксує дисперсію даних.

Вибір конкретної моделі залежить від таких факторів як особливості вхідних даних, наявність в часовому ряді тренду та сезонності та вимоги до прогнозування відносно поставленої задачі. Детальніше про конкретні моделі прогнозування в розділі 1.4.

1.3.3. Перевірка моделі

Перевірка моделі складається з оцінки моделі прогнозування для визначення наскільки добре вона справлятиметься із поставленою задачею. Мається на увазі не просто оцінка «відповідності» моделі історичним даним, а перевірка величини помилок прогнозування на нових, не відомих для моделі даних.

Для перевірки того, наскільки добре модель справляється з новими даними зазвичай використовують підхід розподілу даних, за якого дані поділяються на 2 частини – «підгоночну» вибірку (*fitting segment*) та тестувальну вибірку (*forecasting segment*). При використанні моделей машинного або глибинного навчання «підгоночну» вибірку зазвичай називають навчальною. За такого підходу модель підгоняється до одного сегменту даних, а потім отримана модель використовується для прогнозування даних другого сегменту. Ці прогнози потім порівнюються з реальними спостереженнями сегменту, що допомагає оцінити наскільки добре модель працюватиме з новими даними та якою буде похибка. Саме величина похибки і є показником якості моделі. [5]

Найчастіше використовується середня квадратична помилка (MSE) та середня абсолютна помилка (MAE). Для вимірювання точності часового ряду використовується середня абсолютна відсоткова помилка (MAPE).

Вона вимірюється як відсоток різниці прогнозованих і реальних значень до реальних значень за кожен період. [6]

1.3.4. Застосування та моніторинг роботи моделі

Останнім кроком у прогнозуванні часових рядів є безпосереднє застосування моделі до вхідних даних та отримання результатів. Якщо модель буде застосовуватись не разово, а протягом певного часу, то потрібно подбати про подальший моніторинг роботи моделі. Оцінка роботи моделі має відбуватись з певною періодичністю на постійній основі, аби впевнитись що результат прогнозів лишається задовільним. Природа прогнозування часових рядів є такою, що умови змінюються з часом, і модель, яка добре працювала в минулому, ризикує втратити свою актуальність та точність.[6] Тому моніторинг роботи моделі є невід'ємною частиною дизайну якісної системи для прогнозування.

1.4 Моделі для прогнозування часових рядів

Для прогнозування часових рядів з моменту виникнення цієї проблеми застосовувалась велика кількість різноманітних підходів і моделей. Значна кількість із них покладається на статистичні методи. Проте, із появою та розвитком методів машинного навчання, їх також почали застосовувати для задач прогнозування, причому доволі успішно. Починаючи з 1980 року, багато досліджень було зосереджено на визначенні умов, за яких різні методи працюють найкраще. [7] Досі не визначено однієї «ідеальної» моделі, яка була б кращою за всі інші у 100% завдань. Тому, важливо розуміти особливості кожного підходу хоча б у загальних рисах, аби обрати той що підходить для конкретної задачі та переконатись, що він є дійсно найоптимальнішим.

Загалом наразі використовується три типи моделей для прогнозування часових рядів: класичні моделі, моделі контрольованого машинного навчання та моделі глибинного навчання. [8]

Класичні моделі для прогнозування часових рядів є традиційними статистичними підходами, які широко використовувалися десятиліттями, та незважаючи на появу більш складних алгоритмів машинного навчання, залишаються незамінними. Вони пропонують простоту, інтерпретацію та ефективність, особливо коли маємо справу з невеликими наборами даних або коли потрібна модель, яку можна легко пояснити та зрозуміти. [9] Класичні моделі включають такі як, наприклад, Авторегресія (AR), Ковзне середнє (MA), Авторегресивне ковзне середнє (ARMA), Авторегресивне інтегроване ковзне середнє (ARIMA), Сезонне авторегресивне інтегроване ковзне середнє (SARIMA) та Експоненційне згладжування (ES). Недолік класичних моделей полягає в тому, що вони можуть не працювати належним чином, якщо дані шумні, нелінійні або мають складну динаміку. [10]

Однією з альтернатив класичним методам є **моделі керованого машинного навчання**, які працюють зовсім інакше, ніж класичні моделі. Основна відмінність полягає в тому, що вони поділяють всі змінні на або залежні, або незалежні. Залежні змінні або цільові змінні – це змінні, які хочемо передбачити. Незалежні змінні – це змінні, які допомагають передбачити залежні змінні. [8] У даних часових рядів часто немає незалежних змінних. Проте, досить просто адаптувати їх для використання методів керованого машинного навчання, перетворивши сезонність (на основі, наприклад, позначок часу) у незалежні змінні. До моделей керованого машинного навчання відносять такі як, наприклад, лінійна регресія (linear regression), рандомні ліси (random forests) та XGBoost (eXtreme Gradient Boosting). Методи машинного навчання є потужнішими, гнучкішими та більш адаптивними, ніж класичні методи, але вони також

можуть вимагати більше даних, обчислень та налаштування. Також вони можуть бути не дуже інтерпретованими чи прозорими. [10]

Моделі глибинного навчання це найновіший клас моделей що можна застосовувати для прогнозу часових рядів. Вони ще складніші для сприйняття та освоєння ніж попередні, і вони можуть (але не завжди) давати кращі результати, залежно від даних і специфіки варіанту використання. Серед моделей глибинного навчання для прогнозування часових рядів застосовуються, зокрема, такі як Prophet, DeepAR, також нейронні мережі. Моделі глибинного навчання є найбільш складним класом моделей що застосовуються для прогнозування часових рядів, проте і найбільш потужними серед усіх. Недавні дослідження показали, що нейронна мережа здатна досить добре апроксимувати будь-яку безперервну функцію для прогнозування часових рядів. У той час як класичні методи, такі як ARMA та ARIMA, припускають лінійне співвідношення між входами та виходами, нейронні мережі не зв'язані цим обмеженням. Вони здатні апроксимувати будь-яку нелінійну функцію без попереднього знання про властивості рядів даних. [11]

Розділ 2. Основні теоретичні відомості про нейронні мережі

2.1 Основні відомості про нейронні мережі

Нейронні мережі — це обчислювальні моделі, які імітують біологічні мережі, їх структуру та функціонування, з метою розв'язання різноманітних задач.

Компоненти мережі включають нейрони, з'єднання, ваги, зміщення, функції поширення та правило навчання. Нейрони отримують вхідні дані, керовані пороговими значеннями та функціями активації. Зв'язки включають ваги та зміщення, що регулюють передачу інформації. Навчання,

коригування ваг і зміщень відбувається в три етапи: обчислення вхідних даних, генерація вихідних даних і ітераційне вдосконалення, що підвищує майстерність мережі в різноманітних завданнях.

Глибинне навчання пропонує багато типів нейронних мереж, які можна класифікувати залежно від структури, потоку даних, використовуваних нейронів та їх щільності, шарів та їх глибинних функцій активації тощо. Найкраще у задачах прогнозування часових рядів себе проявили рекурентні мережі (RNN) та їх різновиди — мережі довготривалої короткочасної пам'яті (LSTM) та вентиляльні рекурентні вузли (GRU). Ці моделі революціонізували наш підхід до прогнозування часових рядів, пропонуючи деталізовані та комплексні методи обробки послідовних даних. [12]

2.2. Рекурентні нейронні мережі (RNN)

Рекурентна нейронна мережа — це тип нейронної мережі, де вихідні дані попереднього кроку подаються як вхідні дані для поточного кроку. Звідси і береться назва – подібно до рекурсії, рекурентна нейронна мережа має певну «зацикленість» і використовує власні вихідні дані як вхідні для наступної ітерації. RNN виникли як відповідь на потребу запам'ятовувати та використовувати проміжні результати в процесі прогнозування послідовних даних.

Загальна будова RNN:

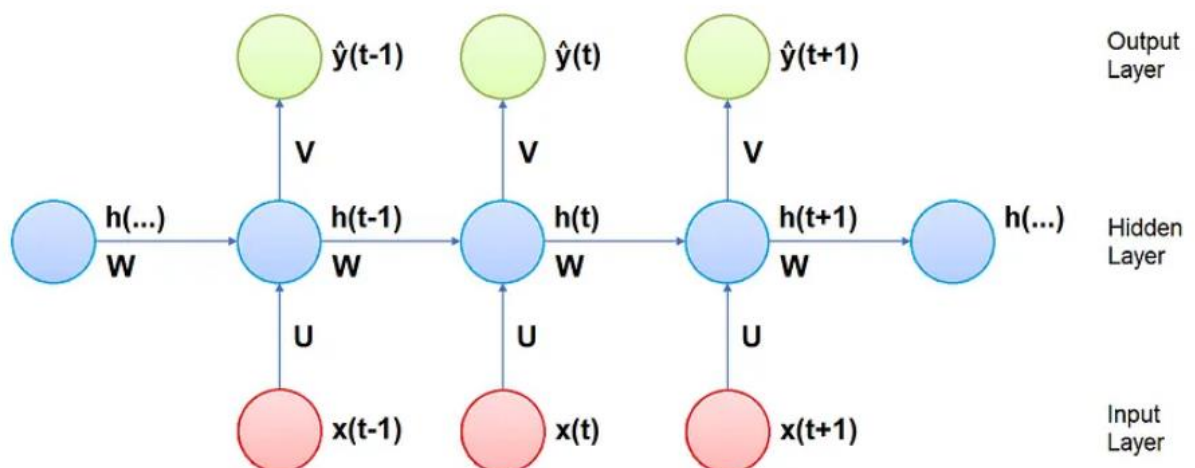
Подібно до класичної нейронної мережі прямого поширення (FNN), проста рекурентна нейронна мережа (RNN) містить три шари: вхідний, прихований і вихідний шар. Що відрізняє RNN від FNN, так це прихований шар. У FNN він тільки передає інформацію вперед до вихідного шару, тоді як у RNN він також передає інформацію назад у себе, утворюючи цикл. Ще одна відмінність полягає у тому, що основною оброблювальною одиницею

у RNN рекурентний вузол, який ще називається рекурентним нейроном. [14] Вузол має здатність підтримувати прихований стан, дозволяючи мережі фіксувати послідовні залежності, запам'ятовуючи попередні вхідні дані під час обробки.

Також, в RNN кожному нейрону призначається фіксований часовий крок і активація нейронів обчислюється в часовому порядку: на будь-якому даному часовому кроці лише нейрони, призначені цьому часовому кроку, обчислюють свою активацію. .

Алгоритм роботи RNN:

Рис. 1 – Розгорнута будова RNN [18]



RNN приймає вхідний вектор X і генерує вихідний вектор y , скануючи дані послідовно зліва направо, з кожним кроком у часі оновлюючи прихований стан і створюючи вихід. В якості вагових коефіцієнтів RNN використовують один і той же набір параметрів U, V, W .

U - ваговий параметр, що регулює з'єднання від вхідного шару X до прихованого шару h
 W - вага, пов'язана зі з'єднанням між прихованими шарами
 V - вага для з'єднання від прихованого шару h до вихідного шару y .
 Спільний доступ до параметрів дозволяє RNN ефективно фіксувати

тимчасові залежності та ефективніше обробляти послідовні дані, зберігаючи інформацію з попереднього введення в поточному прихованому стані.

Прихований стан – вектор $h(t)$ є прихованим станом у момент часу t і є свого роду пам'яттю мережі; він обчислюється на основі поточних вхідних даних та прихованого стану попереднього кроку часу:

$$h(t) = \tanh(W_h(t-1) + U_x(t))$$

Вектор $y^{\wedge}(t)$ — вихід мережі в момент часу t :

$$y^{\wedge}(t) = \text{softmax}(V_s(t)) \text{ [18]}$$

Для навчання у RNN використовується метод зворотного поширення у часі (BPTT), який є вдосконаленою версією звичайного алгоритму зворотного поширення. BPTT рухається не просто назад, а просувається вздовж розгорнутої у часі мережі, обчислюючи зміни ваг на основі похідної помилки відносно цих ваг.

Загалом RNN вирішують багато проблем традиційних моделей машинного навчання для прогнозування часових рядів, проте, у RNN є також один значимий недолік - проблема зникаючого та вибухаючого градієнта. Оскільки градієнт функції втрат поширюється назад у часі під час BPTT, то відбувається багаторазове множення градієнтів на кожному часовому кроці. Як наслідок, він може стати дуже малим (зникнути) або дуже великим (вибухнути). У результаті параметри в прихованих шарах або майже не оновлюються, або це призводить до чисельної нестабільності та хаотичної поведінки. Це призводить до втрати важливої інформації та повільної конвергенції навчання. [16]

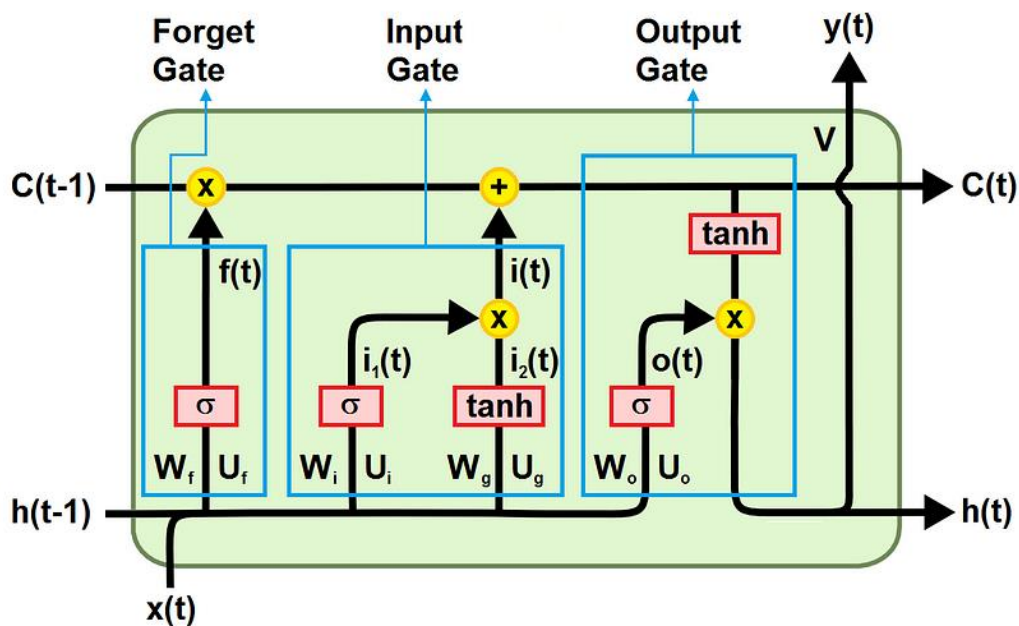
2.3 Мережі довготривалої короткочасної пам'яті (LSTM)

LSTM були розроблені для подолання проблеми зникнення градієнта в стандартній RNN шляхом покращення градієнтного потоку в мережі. [17] Це досягається за допомогою використання замість стандартного прихованого шару вузла LSTM, яка має наступні складові:

- стан вузла, який передає інформацію по всій послідовності та представляє пам'ять мережі
- фільтр забуття, який вирішує, що потрібно зберегти з попередніх часових кроків, а що ні
- вхідний фільтр, який вирішує, яку інформацію необхідно додати з поточного кроку часу
- вихідний фільтр, який визначає значення виходу на поточному кроці часу.

На Рис. 2 нижче проілюстрований принцип роботи вузла LSTM.

Рис. 2 - Вузол LSTM [18]



Подібно до RNN, вагами є параметри U , W , V .
 U – вагова матриця, що регулює з'єднання від вхідного вектору із вузлом LSTM
 часу t

W – вагова матриця, що пов’язує вузол часу t із вузлами часу $t-1$ та $t+1$

V – вагова матриця, пов’язана з вихідним вектором часу t

Матриці W і U поділені на підматриці ($W_f, W_i, W_g, W_o; U_f, U_i, U_g, U_o$), які пов’язані з різними елементами вузла LSTM.

Під час навчання на всіх етапах часу фільтри детермінують, яку інформацію важливо зберегти, а яку забути, і додають її до стану вузла або видаляють із неї. Таким чином LSTM дозволяє зберігати дані від більш ніж одного попереднього кроку, а також відновлювати дані, передані в пам’ять, вирішуючи проблему зникаючого градієнта.

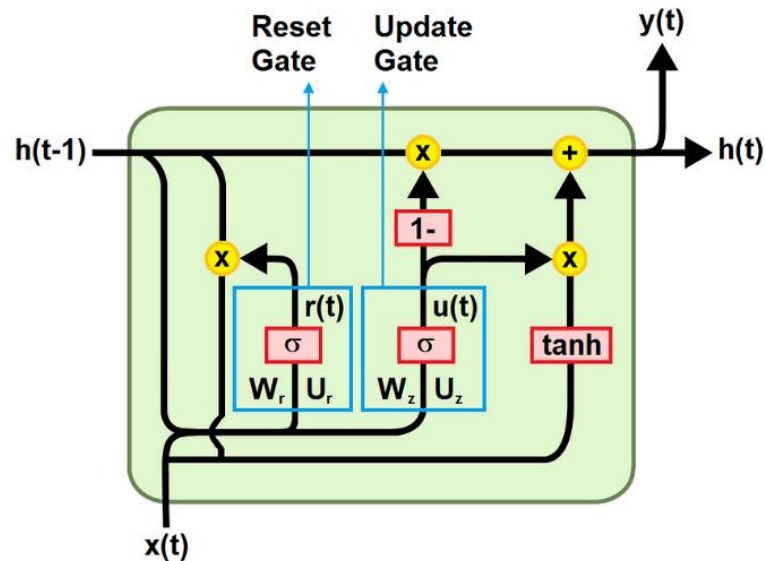
Завдяки своїй специфіці, LSTM використовуються частіше за звичайні RNN та найкраще показують себе в ситуаціях, де необхідно врахувати довгострокові тимчасові залежності, наприклад, для прогнозування погоди або прогнозування моделей споживання електроенергії протягом місяців або років.

2.4 Вентильний рекурентний вузол (GRU)

GRU – тип архітектури рекурентної нейронної мережі (RNN), подібної до LSTM. Як і LSTM, GRU розроблені для моделювання послідовних даних, вибіркового збереження або відкидання інформації з часом, проте має простішу архітектуру і є простішими з точки зору обчислень. Основна відмінність полягає в тому, як вони обробляють стани вузла пам’яті: у той час як LSTM використовує три фільтри (вхідні, вихідні та забуття) щоб підтримувати стан вузла пам’яті окремо від прихованого стану, GRU має два – скидання та оновлення (update and reset gate). [19]

Вузол GRU має наступну структуру, що проілюстрована на Рис. 3:

Рис. 3 – Вузол GRU: [18]



- фільтр скидання, який визначає, скільки інформації з попередніх часових кроків можна забути;
- фільтр оновлення, який вирішує, скільки інформації з попередніх часових кроків потрібно зберегти;
- пам'ять вузла, яка переносить інформацію по всій послідовності та представляє пам'ять мережі.

Вагові коефіцієнти, що використовуються у GRU є аналогічними до інших різновидів рекурентних нейронних мереж і так само є спільними для всієї мережі.

GRU використовуються в подібних контекстах, як LSTM, і є особливо ефективні, коли розмір набору даних менший або коли в пріоритеті ефективність обчислень, оскільки вони менш складні та швидші для навчання. [20]

Розділ 3. Прогнозування даних часового ряду за допомогою нейромережевих моделей

3.1 Датасет

Для задачі прогнозування було обрано дані повітряних тривог України за час повномасштабного вторгнення. Таке рішення було прийнято з двох причин. По-перше, досить важко знайти вільно доступні дані часових рядів, на які б не існувало великої кількості робіт з прогнозування. По-друге, питання прогнозування повітряних тривог є досить цікавим та актуальним на сьогоднішній день. Дані, що використовуються в цій роботі, були взяті з публічного github репозиторію Вадима Клименка [21], що зібрав їх з загально доступних джерел.

Датасет містить два різних набори даних: офіційний та волонтерський. Офіційний походить тільки з офіційних джерел, а саме з телеграм-каналу «Повітряна тривога». Волонтерський набір даних містить також інформацію, зібрану волонтерами в телеграм-каналі «Тривога».

У волонтерському наборі даних для деяких тривог відсутній час закінчення. Ці випадки були оброблені автором так, що час закінчення тривоги було визначено як час початку + 30 хв, а входження даних марковане в спеціальній колонці. Для якісного прогнозування потрібні точні та достовірні дані. З цієї причини було прийняте рішення використовувати для прогнозування тільки офіційний набір.

Перед тим, як використовувати ці дані в дослідженні, потрібно з'ясувати, чи можна вважати ці дані часовим рядом? Відповідь так. Представлені дані містять значення певної змінної, у нашому випадку наявність повітряної тривоги, пов'язаних з певними відмітками у часі, а отже можуть бути представлені у вигляді часового ряду. Чому можуть бути? Бо наразі в даних кожна повітряна тривога є окремим записом, а для того

аби вважатись часовим рядом, точки мають бути рівновіддаленими у часі. Тобто, перед тим як переходити до задачі прогнозування, необхідно провести агрегацію та підготовку набору даних.

3.2 Обрані технології

Для роботи була обрана мова програмування Python, що часто використовується для машинного навчання завдяки своїй високорівневості, що дає змогу зосередитись на важливих речах, не вдаючись до деталей імплементації з точки зору програмування.

У роботі також використовуються певні допоміжні бібліотеки: NumPy, pandas – для роботи з даними у зручному форматі TensorFlow, Keras, Scikit-learn – машинне навчання і нейронні мережі Matplotlib – побудова графіків та візуалізація Statsmodels – статистичні моделі для дослідження даних часового ряду

3.3 Підготовка та дослідження даних часового ряду

Прогнозування буде здійснено для кількості та середньої тривалості тривоги за день по всій Україні. Для цього набір даних необхідно трансформувати: прибрати непотрібні змінні, агрегувати к-ть повітряних тривоги як подобову суму, а тривалість – подобове середнє. Отримуємо набір даних що має 2 змінні, прив'язані до дати: змінну кількості тривоги за добу `air_raids_count` та змінну середньої тривалості тривоги за добу `average_duration`. Нижче для розуміння структури даних наведено перші 5 записів із 778-ми:

date	air_raids_count	average_duration
2022-03-15	46	99.750725
2022-03-16	116	109.094397
2022-03-17	87	49.791762
2022-03-18	113	54.969469
2022-03-19	64	82.297396

Для загального розуміння даних часового ряду їх спершу необхідно візуалізувати. На графіках, наведених нижче (Рис. 4), вісь абсцис – це часова вісь, а вісь ординат – значення відповідної змінної.

Рис. 4



Наступним кроком в аналізі даних часового ряду є дослідження його компонентів, тобто тренду, сезонності та шуму. Результат декомпозиції для кожної із двох змінних наведено нижче (Рис.5 та Рис.6).

Рис. 5 – Декомпозиція часового ряду кількості тривог

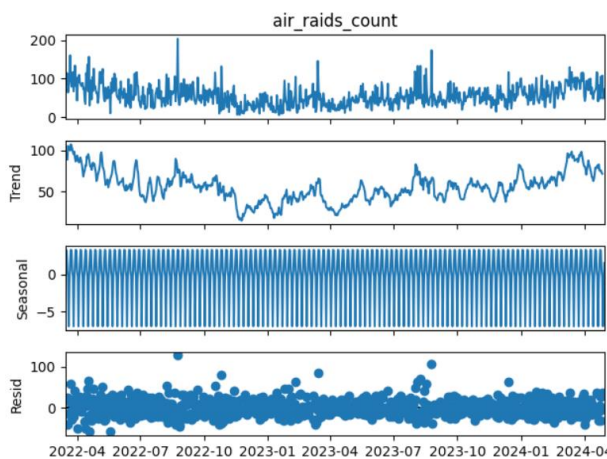
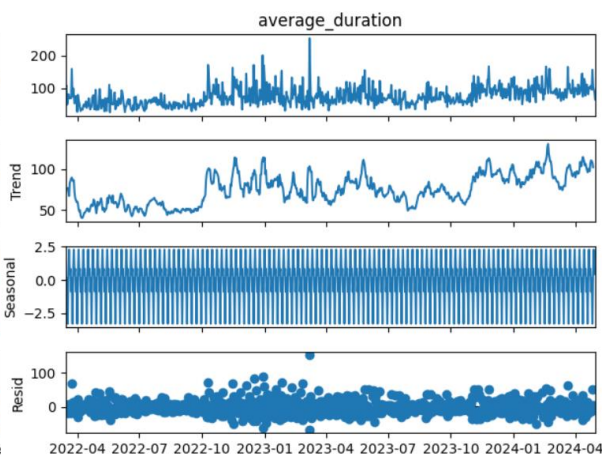


Рис. 6 – Декомпозиція часового ряду тривалості тривог



На основі даних декомпозиції можна зробити кілька висновків. По-перше, шум для обох змінних є досить сильним. Це означає, що передбачити такий часовий ряд буде досить складно і наврядчи вдасться досягти високого результату прогнозування, адже шум – хаотична величина. По-друге, з огляду на сумнівну сезонність, має сенс висунути гіпотезу щодо стаціонарності ряду. Перевірка гіпотези про стаціонарність здійснюється за допомогою тесту Дікі-Фуллера.

Рис. 7 – Тест Дікі-Фуллера для змінної кількості тривог

```
ADF Statistic: -4.003254939258463
p-value: 0.0013935630108979757
Reject the null hypothesis - The time series is stationary
```

Рис. 8 – Тест Дікі-Фуллера для змінної середньої тривалості тривог

```
ADF Statistic: -2.787500087091664
p-value: 0.060081594657884305
Fail to reject the null hypothesis - The time series is non-stationary
```

За результатами тесту (Рис. 7 та Рис. 8) за рівня значущості 0.05 часовий ряд по змінній кількості тривог є стаціонарним, а по змінній середньої тривалості тривог – нестаціонарним, хоча значення p-value і ледь-ледь перетнуло порог значущості.

Якби прогнозування здійснювалось статистичними методами за допомогою класичних методів прогнозування, потрібно було б детальніше розібратися із компонентами ряду та привести його до стаціонарного. Проте, в даному випадку аналіз був проведений суто для розуміння даних, адже нейронні мережі чудово справляються із моделюванням складних зв'язків, в тому числі вони здатні обробляти дані що містять тренд і сезонність.

Натомість, для використання нейромережевих моделей потрібно надати даним певну структуру. Це включає переведення усіх значень змінної в шкалу $[0;1]$, створення лагів, а також розділення вибірки на

навчальну та тестувальну. Лаги тут використовуються на 2 попередні кроки, а дані розбиті по вибірках у співвідношенні 0.7/0.3 – 70% для навчальної вибірки, 30% для тестувальної.

3.3 Створення та тренування моделей

Для побудови всіх трьох моделей - RNN, LSTM та GRU, в даній роботі використовується Python бібліотека TensorFlow.

Для всіх трьох моделей використовувалось 100 проходжень з learning rate 0.01. Для RNN а в якості функції було використано ReLU, для LSTM оптимізатор Adam. Стосовно кількості нейронів у прихованому шарі кожної мережі, то оптимальну визначити не вдалося, бо залежність точності моделі від цього виявилась не сильною, але не стабільною, тобто змінювалась при повторному тренуванні моделі із тими самими параметрами.

Тренування моделей було проведено на попередньо відділеній навчальній вибірці.

3.4 Оцінка моделей

Аби оцінити та порівняти між собою моделі, для кожної було проведено прогнозування на навчальній та тестовій вибірці, та проведена оцінка за допомогою квадратних коренів середньої квадратичної похибки (RMSE) та середня абсолютна відсоткова помилка (MAPE).

Прогнозування для всіх трьох моделей було проведено в кілька ітерацій для кожного набору параметрів. При цьому похибка незначно варіювалось при кожній ітерації незалежно від зміни параметрів. Точність прогнозування варіювалась між моделями незначно, будучи стабільно дещо гіршими для звичайної RNN. Нижче наведено отримані результати оцінки прогнозування на одній із ітерацій (Рис. 9).

Рис. 9 – Оцінка результатів прогнозування даних часового ряду нейромережевими алгоритмами

Evaluation of the RNN model:

17/17 ————— 0s 2ms/step

8/8 ————— 0s 2ms/step

Train Score RNN (Count): 0.13 RMSE, 84.12% MAPE

Test Score RNN (Count): 0.14 RMSE, 35.83% MAPE

Train Score RNN (Duration): 0.12 RMSE, 97.02% MAPE

Test Score RNN (Duration): 0.12 RMSE, 32.26% MAPE

Evaluation of the LSTM model:

17/17 ————— 1s 16ms/step

8/8 ————— 0s 2ms/step

Train Score LSTM (Count): 0.11 RMSE, 75.57% MAPE

Test Score LSTM (Count): 0.12 RMSE, 36.14% MAPE

Train Score LSTM (Duration): 0.10 RMSE, 86.15% MAPE

Test Score LSTM (Duration): 0.11 RMSE, 28.70% MAPE

Evaluation of the GRU model:

17/17 ————— 0s 3ms/step

8/8 ————— 0s 2ms/step

Train Score GRU (Count): 0.12 RMSE, 74.81% MAPE

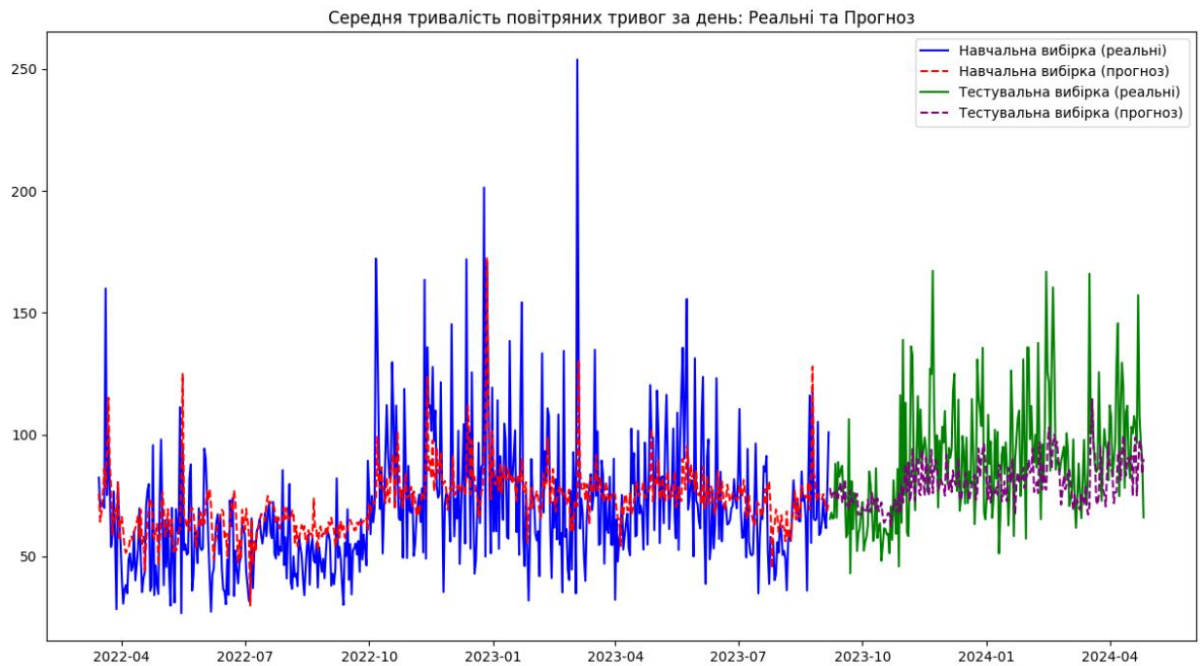
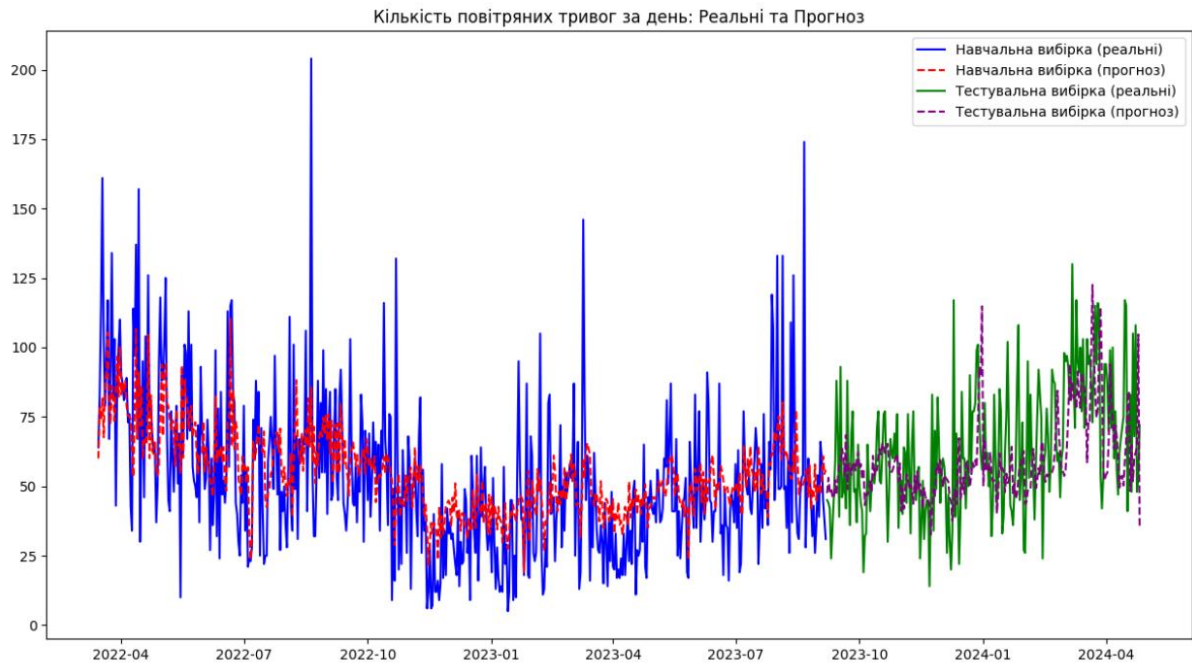
Test Score GRU (Count): 0.12 RMSE, 34.23% MAPE

Train Score GRU (Duration): 0.11 RMSE, 79.56% MAPE

Test Score GRU (Duration): 0.11 RMSE, 27.42% MAPE

Більш наглядними для розуміння загальної точності моделей тут є значення MAPE. MAPE показує середнє відсоткове відхилення прогнозованого значення від реального спостереження. Тут в ході прогнозування на тестувальній вибірці було отримано похибку близько 35% для змінної кількості тривог та 30% для змінної їх середньої тривалості. Традиційно точність MAPE 20-50% вважається не високою, але прийнятною [22]. Враховуючи досить сильну зашумленість даного часового ряду, яку було встановлено на етапі декомпозиції, цей результат є досить таки хорошим. Нижче наведено візуалізацію результатів прогнозування для моделі LSTM (Рис.10).

Рис. 10 – Візуалізація прогнозування LSTM



На графіку можна побачити, що нейронна мережа на достатньому рівні вловлює основні короткотривалі та довготривалі закономірності часового ряду.

Висновки

У даній роботі проведено дослідження прогнозування часових рядів нейромережевими алгоритмами. В якості теоретичного підґрунтя було розглянуто основи часових рядів, їх обробки і прогнозування, а також типи рекурентних нейронних мереж що використовуються для задач прогнозування послідовних даних та їх будову. Задля перевірки роботи нейромережових алгоритмів на практиці було обрано дані повітряних тривог в Україні за час повномасштабного вторгнення, агреговані подобою. Для цих даних було проведено прогнозування за допомогою 3 різних моделей: звичайна RNN, LSTM та GRU. Точність отриманих результатів для всіх трьох моделей була задовільною, проте не дуже високою, що було виявлено як наслідок особливостей вхідних даних. Найкращий результат був отриманий при використанні моделей LSTM та GRU.

На основі отриманих результатів можна зробити висновок, що рекурентні нейронні мережі є потужним інструментом прогнозування часових рядів, хоча і не ідеальним. Вони показують задовільний результат навіть на складних даних із нелінійними зв'язками, отже без проблем можуть використовуватись для вирішення реальних задач прогнозування.

Список використаної літератури

1. "Аналіз та прогнозування часових рядів" [Електронний ресурс] / Drukarnia – Режим доступу до ресурсу: <https://drukarnia.com.ua/articles/analiz-ta-prognozuvannya-chasovikh-ryadiv-H6gg0#heading-2-53>.
2. Santra, Ritu. "Tests for Stationarity in Time Series: Dickey-Fuller Test, Augmented Dickey-Fuller (ADF) Test" [Електронний ресурс] / Medium – Режим доступу до ресурсу: <https://medium.com/@ritusantra/tests-for-stationarity-in-time-series-dickey-fuller-test-augmented-dickey-fuller-adf-test-d2e92e214360>.
3. Tiwari, Ajay. "Build Foundation for Time Series Forecasting" [Електронний ресурс] / Towards Data Science – Режим доступу до ресурсу: <https://towardsdatascience.com/time-series-forecasting-968192b3781a>.
4. "What is Time Series Analysis?" [Електронний ресурс] / Spotfire – Режим доступу до ресурсу: <https://www.spotfire.com/glossary/what-is-time-series-analysis>.
5. Montgomery DC, Jennings CL, Kulahci M. Introduction to Time Series Analysis and Forecasting. Wiley-Interscience; 2008.
6. Pathak, Puja P. "Time Series Forecasting: A Complete Guide" [Електронний ресурс] / Medium – Режим доступу до ресурсу: <https://medium.com/analytics-vidhya/time-series-forecasting-a-complete-guide-d963142da33f>.
7. Remus, W., O'Connor, M. Neural Networks for Time-Series Forecasting. Springer, Boston, MA; 2001.
8. Korstanje, Joos. "Select Model for Time Series Prediction Task" [Електронний ресурс] / Neptune – Режим доступу до ресурсу: <https://neptune.ai/blog/select-model-for-time-series-prediction-task>.

9. Ali. "Introduction to Classical Time Series Models: The Cornerstones of Forecasting" [Электронный ресурс] / Medium – Режим доступа до ресурсу: <https://medium.com/@Alidotab/introduction-to-classical-time-series-models-the-cornerstones-of-forecasting-part-0-1db85fe60921>.
10. "What Are the Most Effective Methods for Time Series Forecasting?" [Электронный ресурс] / LinkedIn – Режим доступа до ресурсу: <https://www.linkedin.com/advice/1/what-most-effective-methods-time-series-forecasting>.
11. "What is Time Series Forecasting?" [Электронный ресурс] / Timescale – Режим доступа до ресурсу: <https://www.timescale.com/blog/what-is-time-series-forecasting/>.
12. "Types of Neural Networks" [Электронный ресурс] / Great Learning – Режим доступа до ресурсу: <https://www.mygreatlearning.com/blog/types-of-neural-networks/>.
13. Kalita, Debrasish. "A Brief Overview of Recurrent Neural Networks (RNN)" [Электронный ресурс] / Analytics Vidhya – Режим доступа до ресурсу: <https://www.analyticsvidhya.com/blog/2022/03/a-brief-overview-of-recurrent-neural-networks-rnn/>.
14. "Introduction to Recurrent Neural Network (RNN)" [Электронный ресурс] / GeeksforGeeks – Режим доступа до ресурсу: <https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/>.
15. Poudel, Sushmita. "Recurrent Neural Network (RNN) Architecture Explained" [Электронный ресурс] / Medium – Режим доступа до ресурсу: <https://medium.com/@poudelsushmita878/recurrent-neural-network-rnn-architecture-explained-1d69560541ef>.
16. Dr Barak Or. "The Exploding and Vanishing Gradients Problem in Time Series" [Электронный ресурс] / Metaor – Режим доступа до ресурсу: <https://medium.com/metaor-artificial-intelligence/the-exploding-and-vanishing-gradients-problem-in-time-series-6b87d558d22>.

17. "LSTM and the Vanishing Gradient Problem" [Электронный ресурс] / Baeldung – Режим доступа до ресурсу: <https://www.baeldung.com/cs/lstm-vanishing-gradient-prevention>.
18. Marco Del Pra. "Time Series Forecasting with Deep Learning and Attention Mechanism" [Электронный ресурс] / Towards Data Science – Режим доступа до ресурсу: <https://towardsdatascience.com/time-series-forecasting-with-deep-learning-and-attention-mechanism-2d001fc871fc>.
19. Anishnama. "Understanding Gated Recurrent Unit (GRU) in Deep Learning" [Электронный ресурс] / Medium – Режим доступа до ресурсу: <https://medium.com/@anishnama20/understanding-gated-recurrent-unit-gru-in-deep-learning-2e54923f3e2>.
20. Gomedede, Everton. "Exploring the Efficacy and Efficiency of Gated Recurrent Unit (GRU) Layers in Sequential Data" [Электронный ресурс] / Medium – Режим доступа до ресурсу: <https://medium.com/the-modern-scientist/exploring-the-efficacy-and-efficiency-of-gated-recurrent-unit-gru-layers-in-sequential-data-1960e45687de>.
21. Klymenko, Vadym. Ukrainian Air Raid Sirens Dataset [Электронный ресурс] / GitHub – Режим доступа до ресурсу: <https://github.com/Vadimkin/ukrainian-air-raid-sirens-dataset/tree/main>.
22. Lewis C. D. Industrial and business forecasting methods: A practical guide to exponential smoothing and curve fitting. London: Butterworth Scientific, 1982.