МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В.Н.Каразіна
Факультет математики і інформатики
Кафедра теоретичної та прикладної інформатики

Кваліфікаційна робота

магістр

на тему ”Вибір властивостей рекурентною нейронною мережею як метод аналізу ефектів властивостей середовища”

Виконав: студент 2 курсу, групи МФ-61
спеціальність 122 Комп’ютерні науки
освітньо-наукова програма «Інформатика»

Котенко Дмитро Анатолійович
(прізвище та ініціали)
Керівник Меняйлов Є.С.
(прізвище та ініціали)
Рецензент Щуцкий Д.О.
(прізвище та ініціали)

Харків – 2024 року

ВСТУП

1.1 Мета і завдання дослідження

Мета дослідження полягає в розробці алгоритму аналізу даних на основі штучного інтелекту, який відсортовує атрибути досліджуваного набору даних за їхнім впливом на цільовий часовий ряд. В ході роботи слід також визначити метрики за якими буде отримуватись рівень впливу на часовий ряд.

Вхідні дані - часовий ряд проблеми яка досліджується та властивості середовища (аттрибути) які експерти в цій галузі виділили як каузальні до проблеми. Вихідні - відсортовані властивості за тим чи мали вони вплив на проблему, чи ні.

1.2 Актуальність теми

За результатами глобального опитування проведеного консалтинговою компанією McKinsey & Company від 2022 року [1] ми можемо побачити що кількість компаній які використовують ШІ у своїй роботі збільшилась вдвічі в порівнянні з 2017 роком. При цьому прогнозується що до 2040 року цей відсоток досягне 38.4%.

І хоч наразі спостерігається стрімке зростання адаптації саме генеративних систем таких як копайлот які мають за мету автоматизування процесів, великий вплив саме у прогнозах які будують консалтингові компанії йде на те що аналіз даних на основі штучного інтелекту буде все частіше застосовуватись компаніями будівельними та інжиніринговими компаніями.

Також з опитування проведеного MIT [2] ми бачимо тенденцію до того що бізнеси все частіше відмовляються від послуг дата аналітиків на користь програмних продуктів які роблять їх роботу. Аналізу даних на основі штучного інтелекту є частиною таких програмних продуктів.

1.3 Методи дослідження

Для початку в ході роботи висуваються гіпотези про поведінку алгоритму і його властивості. Через природу задачі і алгоритму в ході дослідження перевіряються ці гіпотези, висуваю статистичні гіпотези і статистичні висновки які необхідно перевірити. Статистичні гіпотези стосуються кількості помилок у моделі прогнозування при зміні атрибутів які мають зв’язок з цільовим рядом на випадкові величини.

Для кожної з гіпотез ставиться умова яку необхідно дослідити, шукається реальна статистика і формується задача на статистиці така щоб репрезентувала гіпотезу. Таким чином ми маємо реальну статистику, з вже доведеними або очевидними явищами які і перевіряються.

Для гіпотези і статистики розробляється програма обробки даних, нейронна мережа, реалізація моделі алгоритму, обробка метрик і візуалізація графіків метрик.

Метрики які використовуються в дослідженні:

- Середня абсолютна помилка (Mean Absolute Error, MAE)
- Корінь середньої квадратичної помилки (Root Mean Squared Error, RMSE)

1.4 Практичне значення одержаних результатів

Результатом дослідження є гібридне рішення зниження розмірності з використанням нейронної мережі.

Також за результатами проведеного дослідження графіки та моделі які отримуються при роботі програми можуть бути використані як результати розвідувального аналіз (Exploratory data analysis) і на їх основі в подальшому можна формувати нові гіпотези.

1.5 Публікації

Робота була представлена на XIX Міжнародну науково-практичну конференцію "Сучасні проблеми математики та її застосування у природничих науках та інформаційних технологіях", що відбулась 10-11 травня 2024 року.

Робота складається зі вступу, дев’яти глав, загальних висновків, списку використаної літератури (15), додатків (14). Зміст роботи висвітлено на 37 сторінках основного тексту і містить 5 таблиць і 7 рисунків.

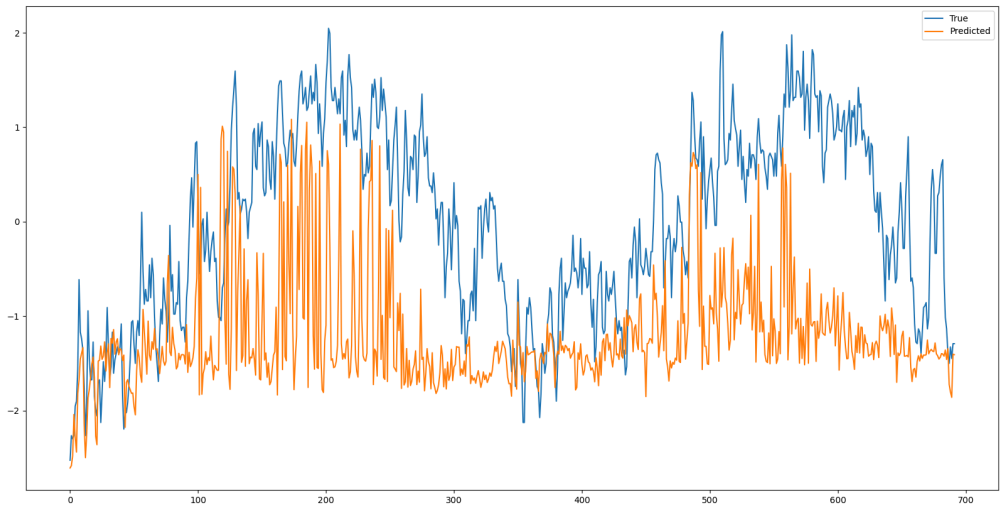
ОСНОВНА ЧАСТИНА

2.1 Постановка задачі

Дослідження полягає в розробці алгоритму аналізу даних на основі штучного інтелекту, який відсортовує атрибути досліджуваного набору даних за їхнім впливом на цільовий часовий ряд. В ході роботи слід також визначити метрики за якими буде отримуватись рівень впливу на часовий ряд.

Вхідні дані - часовий ряд проблеми яка досліджується та властивості середовища (аттрибути) які експерти в цій галузі виділили як каузальні до проблеми. Вихідні - відсортовані властивості за тим чи мали вони вплив на проблему, чи ні.

Основна ідея полягає в зниженні якості прогнозування моделі через заміну властивості, яка має вплив на прогнозування, на випадкову величину.



Мал. 2.1. Підміна атрибута який має вплив на прогнозування випадковою величиною.

Тож ціль сформульована до трьох гіпотез які необхідно перевірити за допомогою алгоритму:

- вплив властивості на передбачення є і може бути виміряний.

- можна використовувати результати отримані алгоритмом для отримання нових гіпотез про дані чи тренда.
- алгоритм може працювати і з великою розмірністю атрибутів.

За для перевірки гіпотез було проведено три експерименти на трьох різних даних статистики.

1. Експеримент 1. Перевірка тези про вплив властивості на передбачення за допомогою статистики погоди. факти які перевіряються: глибина снігу не впливає на середньодобову температуру, в той самий час атмосферний тиск впливає на температуру. Відповідно ціль перевірити що заміна реальних даних на випадкову величину (0) для тих атрибутів які не мають вплив - графік не змінюється, для тих які мають вплив - графік спотворений.
2. Експеримент 2. Досліджувана гіпотеза: Твіти Ілона Маска негативні та позитивні напряму впливають на ціну акцій починаючи з січня 2022 (коли почались чутки про купівлю ним твіттера). Усього 10 атрибутів без цільового атрибута (тобто у разі використання обгорткового метода складність 2^10). У ході експеримента перевіряється чи зможуть дані показати залежність з гіпотези, перевірка якості алгоритму на повній множині і підмножинах, перевірити можливість нових гіпотез чи трендів.
3. Експеримент 3. Стрестест. перевірка роботи алгоритму з великою кількістю атрибутів. Прогнозування мавп’ячої віспи на середовищі утвореному міжнародними оцінками здоров’я в країнах світу. Цільовий часовий ряд - динаміка мавп’ячої віспи згрупована по країнах. Середовище - оцінки якості життя та послуг World Health Statistics 2020 згруповані по країнах. Після етапи підготовки даних маємо 39 країн і 100 атрибутів. Складність обгортково алгоритму 2^100 підмножин які необхідно оцінити.

2.2 Чи можна використовувати нейронну мережу як модель при досліджені?

Рекурентні нейронні мережі з довгою короткочасною пам'яттю (LSTM)[3] є надзвичайно потужною моделлю глибокого навчання, яка може легко моделювати проблеми з декількома вхідними змінними часових рядів даних. Вона має додаткову перевагу в прогнозуванні часових рядів, де класичні лінійні методи стикаються з труднощами в адаптації до багатовимірних або множинних вхідних даних. LSTM має багато переваг над RNN, оскільки він може впоратися з втратою пам'яті і, таким чином, запобігти затуханню і передчасному зникненню градієнтного спаду нейронних мереж. Крім того, для прогнозування часових рядів він може автоматично навчатися на основі даних, що залежать від часу, і може автоматично обробляти такі часові явища, як тренди та сезонність.

Саме тренди та сезонність були основною проблемою чому прогнозування часових рядів могло вважатись надійним виключно за їх відсутності і потребувало приведення до стаціонарності часового ряду як можна побачити в роботі “Моделювання та прогнозування часових рядів трендів за допомогою нейронних мереж”[4].

У роботі “Прогнозування фінансових та нестаціонарних часових рядів за допомогою рекурентної нейронної мережі LSTM для короткого та довгострокового періоду”[5] рекурентна нейронна мережа на основі довгострокової та короткострокової пам'яті (LSTM) використовується для короткострокового та довгострокового прогнозування для різних часових рядів. Дослідження проведено на чотирьох різних часових рядах даних, а саме: Mackey Glass, DJIA, обмінний курс індійської валюти та японська валютна біржа. Виявлено, що запропонований метод здатен досягти значно кращих результатів порівняно з іншими сучасними алгоритмами, описаними в літературі, такими як нейронні мережі, ELM, регресія на основі опорних векторів.

Із результатів дослідження виходить що завдяки шарам LSTM досягається найкраща ефективність прогнозування на чотирьох різних наборах даних реальних часових рядів.

Таким чином є достатня наукова база вважати що для нестаціонарних даних з використанням нейронні мережі з довгою короткочасною пам'яттю (LSTM) ми отримаємо якісні моделі.

2.3 Як визначається рівень впливу на часовий ряд?

Рівень впливу атрибуту або ж Variable Importance це метрика збільшення помилки передбачення моделі після того, як ми підмінили значення атрибуту, що порушує зв'язок між ознакою та справжнім результатом. Також ця метрика відома як Permutation Feature Importance [6].

Концепція дуже проста: Ми вимірюємо важливість властивості, обчислюючи збільшення помилки прогнозування моделі після перестановки значень цієї властивості. Властивість є «важливою», якщо перемішування її значень збільшує похибку моделі, оскільки в цьому випадку модель покладається на цю властивість для прогнозування. Ознака є «неважливою», якщо перестановка її значень залишає похибку моделі незмінною, тому що в цьому випадку модель ігнорувала цю ознаку при прогнозуванні. Вимірювання Permutation Feature Importance було введено Брейманом [7] для випадкових лісів. На основі цієї ідеї Фішер, Рудін та Домінічі [8] запропонували модельно-діагностичну версію важливості ознаки і назвали її модельною залежністю.

Помилки які використовуються як метрики у дослідженнях і при формуванні результатів це:

- Середня абсолютна помилка (Mean Absolute Error, MAE)

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- Корінь середньої квадратичної помилки (Root Mean Squared Error, RMSE)

$$\sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

2.4 Розгорнутий огляд сучасного стану справ в області

Попереднім технічним рішенням схожим за метою до теми дослідження є зниження розмірності, а сама обрання ознак. Методиками обрання ознак яких є альтернативами до запропонованого мною алгоритму є:

2.4.1 Обгорткові методи

Обгорткові методи, які утворюють підмножини для цих підмножин встановлюється відповідний бал шляхом використання передбачуваної моделі. Оскільки обгортковий моделі для кожної підмоделі тренують нову модель передбачення, вони є дуже напруженими, але зазвичай пропонують множину ознак із найкращою продуктивністю для того окремого типу моделі.

Поширені стратегії в рамках методів обгортки включають в себе наступні:

Прямий відбір:

- Початок з нуля: Почніть з порожньої множини атрибутів та ітеративно додають по одному атрибуту за раз.
- Оцінка моделі: На кожному кроці тренуйте та оцінюйте модель машинного навчання, використовуючи вибрані ознаки.
- Критерій зупинки: Продовжуйте до тих пір, поки не буде досягнуто попередньо визначеного критерію зупинки, наприклад, максимальної кількості ознак або значного падіння продуктивності.

Виключення у зворотному порядку:

- Почати з усього: Почати з усіх доступних функцій.
- Ітеративне видалення: На кожній ітерації видаляйте найменш важливу ознаку та оцінюйте модель.
- Критерій зупинки: Продовжуйте, доки не буде виконано умову зупинки.

Рекурсивне виключення ознак (RFE):

- Ранжування ознак: Почніть з усіх ознак і проранжуйте їх на основі їхньої важливості або внеску в модель. Важливість визначається за допомогою індекса Джині вбудованого в моделі дерев, такі як Random Forest Classifier.
- Ітеративне видалення: На кожній ітерації видаляйте найменш важливі ознаки.
- Критерій зупинки: Продовжуйте, доки не буде досягнуто бажаної кількості ознак.

Вичерпний пошук:

- Вивчення всіх можливостей: Оцінити всі можливі комбінації атрибутів, що гарантує знаходження найкращої підмножини для продуктивності моделі.
- Обчислювальні витрати: Це може бути пов'язано з великими обчислювальними витратами, особливо при великій кількості ознак.

2.4.2 Фільтрові методи

Фільтрові методи використовують міру на основі якої відбувається фільтрація, такою мірою можуть виступати коефіцієнти кореляції або коефіцієнти взаємної інформації.

Плюси використання фільтрових методів як кореляція:

- Кореляція може продемонструвати наявність або відсутність зв'язку між двома факторами, тому добре підходить для визначення сфер, в

яких можна провести експериментальне дослідження і показати подальші результати.

- Швидкість обчислення

Мінуси використання фільтрових методів як кореляція:

- У кореляційному дослідженні не можна встановити причинно-наслідковий зв'язок, оскільки немає впевненості, що одна змінна спричинила іншу, це може бути одна або інша змінна, або навіть невідома змінна, яка спричиняє кореляцію.

2.4.3 Вбудовані методи

Вбудовані методи, що використовують обрання ознак як процес побудови моделі. Прикладом цього підходу є метод побудови лінійних моделей LASSO.
Як одразу можна побачити основними мінусами цих підходів є або їх жадібність як у випадку з обгорткою ми методами, так і їх націленість більше на швидкість як за фільтром методами та вбудованими методами.

2.5 Вибір технологій

Основною мовою розробки було обрано Python з оригінальним середовищем, та з середовищем Anaconda з розширенням для роботи з графічним процесором та RT ядрами за для прискорення процесу навчання моделей передбачення. Завдяки простоті та широкому використанню, Python дозволяє розробникам швидко створювати прототипи та вирішувати різноманітні завдання. Python підтримує об'єктно-орієнтоване програмування (ООП), що дозволяє розробникам створювати модульний та розширюваний код. Python використовує динамічну типізацію, що означає, що ви можете присвоювати змінним будь-який тип даних, і тип даних визначається автоматично в ході виконання програми. Код на Python може бути легко

перенесений між різними платформами, оскільки він є інтерпретованою мовою, а не компільованою. Python має величезне співтовариство розробників, яке розробляє і підтримує безліч корисних бібліотек і фреймворків, що значно спрощує розробку програм.

Бібліотека Pandas. Потужний інструмент для аналізу та обробки даних у мові програмування Python. Вона надає широкий набір функцій і інструментів для роботи з даними у вигляді табличних структур, що дозволяє легко виконувати операції з даними, такі як завантаження, фільтрація, агрегація, об'єднання, аналіз та візуалізація.

Загалом, бібліотека Pandas є незамінним інструментом для роботи з даними в Python, особливо для аналізу та обробки даних у табличному форматі.

Бібліотека NumPy (Numerical Python). Одна з найпопулярніших бібліотек для обчислення в мові програмування Python. Вона надає потужні інструменти для роботи з масивами даних, векторизованими операціями та математичними функціями, що робить її ідеальним інструментом для наукових обчислень та обробки даних.

Бібліотека NumPy є ключовою складовою екосистеми Python для наукових обчислень та обробки даних і широко використовується у багатьох наукових, інженерних та фінансових застосуваннях.

Бібліотека TensorFlow. Це відкрите програмне забезпечення для чисельних обчислень, що використовується для створення та навчання моделей глибокого навчання. Вона розроблена компанією Google та широко використовується у наукових дослідженнях та виробничих застосуваннях.

TensorFlow використовує графову модель обчислень, де ви визначаєте структуру обчислень за допомогою графа, а потім виконуєте обчислення у цьому графі за допомогою сесій.

TensorFlow має велику та активну спільноту розробників, що розробляють різноманітні ресурси, включаючи документацію, онлайн-курси, блоги та інше.

TensorFlow надає можливість використовувати графові обчислення для розпаралелювання та оптимізації обчислень, що робить його дуже продуктивним для тренування складних моделей на великих обсягах даних.

TensorFlow є однією з найпопулярніших бібліотек для глибокого навчання та штучного інтелекту, і використовується в багатьох великих проектах та дослідницьких роботах.

2.6 Опис алгоритму

Крок 1. підготовка даних

Підготовка типова для аналізу даних:

Збір даних

Дані збираються з різних джерел. Вимоги можуть бути передані аналітиками зберігачам даних; наприклад, персонал інформаційних технологій в організації. Дані також можуть бути зібрані з датчиків у навколишньому середовищі, включаючи камери дорожнього руху, супутники, пристрої запису тощо. Їх також можна отримати за допомогою інтерв'ю, завантаження з онлайн-джерел або читання документації.

Обробка даних

Фази циклу розвідки, які використовуються для перетворення необробленої інформації в оперативну інформацію або знання, концептуально подібні до фаз аналізу даних. Дані, отримані спочатку, повинні бути оброблені або організовані для аналізу. Наприклад, це може передбачати розміщення даних у рядках і стовпцях у форматі таблиці (відомих як структуровані дані) для подальшого аналізу, часто за допомогою електронних таблиць або статистичного програмного забезпечення.

Очищення даних

Після обробки та впорядкування дані можуть бути неповними, містити дублікати або містити помилки. Потреба в очищенні даних виникне через проблеми в способі введення та збереження даних. Очищення даних — це процес запобігання та виправлення цих помилок. Загальні завдання включають зіставлення записів, виявлення неточності даних, загальну якість наявних даних, дедуплікацію та сегментацію стовпців.

Наступні кроки є процесом моделювання та створення моделі.

Крок 2. Створення моделі нейронної мережі

На цьому етапі створюється і тренується модель нейронної мережі яка має можливість передбачати цільовий ряд. Акцент саме робиться на якості отриманої моделі оскільки як недонавчена модель не буде мати властивостей прогназування, так і перенавчена модель буде насправді фокусуватись на випадкову похибку або шум, замість взаємозв'язку, що лежить в основі даних.

Задля досягнення цих цілей алгоритм немає обмежень на архітектуру нейронної мережі і кількість її гіпер параметрів.

В експерементах наведених нижче виконується тренування мережі зі слоями LSTM.

Крок 3. Оцінка атрибутів.

Оцінка впливу атрибутів на цільовий часовий ряд (середня температура повітря за день спостереження) проводиться заміною реальних даних атрибуту, на випадкові. В даному випадку для кожного атрибуту окрім цільового проводиться передбачення на даних з заміненими даних атрибуту на випадкові значення з метою зменшення якості прогнозування моделі та збір інформації по помилкам отриманих при такій заміні.

Код процесу оцінки

```
def exp_variables_importance_test(model, dataframe, target_column):
    new_df_values = dataframe.values
    new_df_values = new_df_values.astype('float32')
    # normalize features
    scaler = StandardScaler()
    scaler.fit(new_df_values)

    columns_errors = {}
    columns_set = dataframe.columns
    columns_set = columns_set.drop([target_column])
```

```
    for column in columns_set:
        concrete_test_df = dataframe.copy()
        # change column values to random number
        column_max = concrete_test_df[column].max()
        column_min = concrete_test_df[column].min()
        # random values between min and max
        concrete_test_df[column] = np.random.uniform(column_min, column_max,
len(concrete_test_df))

        new_df_values = concrete_test_df.values
        new_df_values = new_df_values.astype('float32')
        scaled = scaler.transform(new_df_values)
        testX, testY = scaled[:, 1:], scaled[:, 0]
        testX = np.reshape(testX, (testX.shape[0], testX.shape[1], 1))
        predictions = model.predict(testX)

        # get MAE, RMSE for predictions
        mae = np.mean(np.abs(predictions - testY))
        rmse = np.sqrt(np.mean((predictions - testY) ** 2))
        columns_errors[column] = {'mae': mae, 'rmse': rmse}
        # draw test and predictions
        plt.figure(figsize=(20, 10))
        plt.plot(testY, label='True')
        plt.plot(predictions, label='Predicted')
        plt.legend()
        plt.savefig(f'env_predictions_{column}.png')

return columns_errors
```

Крок 4. Робота з підмножинами

Потенціал перенавчання залежить не лише від кількості параметрів та даних, але й від відповідності структури моделі формі даних, та величини похибки моделі в порівнянні з очікуваним рівнем шуму або похибки в даних. Тому задля вирішення проблеми перенавчання можна застосовувати цей крок фокусуючись на підмножинах і повторюючи кроки два та три на підмножинах.

2.7 Експеримент 1.

Метою експеримента є перевірити на практичному прикладі гіпотезу про те що при наявності у нас нейронної мережі яка виконує задачу регресії і має гарні метрики якості, при зміні у атрибуту який має вплив на часовий ряд буде суттєво зменшуватись якість передбачень які робить нейронна мережа.

Досліджуваним часовим рядом було обрано статистику погоди у Лондоні в проміжку від 1970 року і до сьогодення [9]. Набір даних складається з таких атрибутів:

date - дата запису спостереження;
cloud_cover - рівень хмарності в день спостереження;
sunshine - рівень сонячної активності в день спостереження;
global_radiation - радіаційний рівень у день спостереження;
max_temp - максимальна температура повітря у день спостереження;
mean_temp - середня температура повітря за день спостереження;
min_temp - мінімальна температура повітря за день спостереження;
precipitation - кількість опадів в день спостереження;
pressure - атмосферний тиск у день спостереження;
snow_depth - товщина снігу в день спостереження;

Додаток 1. Приклад початкових даних з експерименту 1.

Цільовим атрибутом який буде передбачати нейронна мережа є середньодобова температура. З цих даних можемо одразу зробити такі твердження:

- Глибина снігу не впливає на середньодобову температуру, відповідно є очікування що при зміні значень цього атрибуту на випадкові ми не побачимо значних змін в прогнозуванні (більших ніж похибка, або зимній період).

- Атмосферний тиск має зворотну залежність між температурою повітря та атмосферним тиском: чим вище температура, тим тиск менший і навпаки: чим температура нижча, тим тиск більший. Через це очікую що при зміні значень цього атрибуту ми побачимо спотворення графіку передбачення температури.
- Дата має вплив на передбачення оскільки цей параметр висвітлює сезонні явища які приховані за нею. Через це очікую що при зміні значень цього атрибуту ми побачимо спотворення графіку передбачення температури.

Крок 1. Підготовка даних.

Дані вже достатньо підготовлені, тому етап збору даних для нас не актуальний. Але все ж з атрибутів були видалені мінімальна та максимальну температура повітря за день, оскільки інакше середньодобова температура буде в більшості визначатись за цими параметрами і зіпсує експеримент. Також задля прискорення тренування нейронної мережі було вирішено зменшити розмір дослідження до 5% відсотків від початкового (з 15341 днів до 767 днів).

Крок 2. Тренування моделі нейронної мережі.

Для цього експерименту було вирішено проводити тестування виключно на повному наборі даних. Нейронна модель побудована за допомогою компонентів пакету Tensorflow має сиквенційний вигляд і складається з шару 50 LSTM нейронів та одного вихідного Dense шару. Функція втрат - Середня абсолютна помилка (Mean Absolute Error, MAE). Метод оптимізації - ADAM. Модель тренувалась на тестових даних з кількість епох 200.

Модель була розділена на тренувальну і тестувальну добірку у співвідношенні 90 до 10 для того щоб система мала більше можливості визначитись з сезонними явищами.

Результати тренування моделі:

MAE: 1.1453115940093994

RMSE: 1.3538013696670532

Train MAE: 1.1511173248291016

Train RMSE: 1.4148422479629517

Крок 3. Оцінка атрибутів.

Оцінка впливу атрибутів на цільовий часовий ряд (середня температура повітря за день спостереження) проводиться заміною реальних даних атрибуту, на випадкові. В даному випадку для кожного атрибуту окрім цільового проводиться передбачення на даних з заміненими даних атрибуту на 0 та збір інформації по помилкам отриманих при такій заміні.

Результати відсортовані за кількістю отриманих помилок:

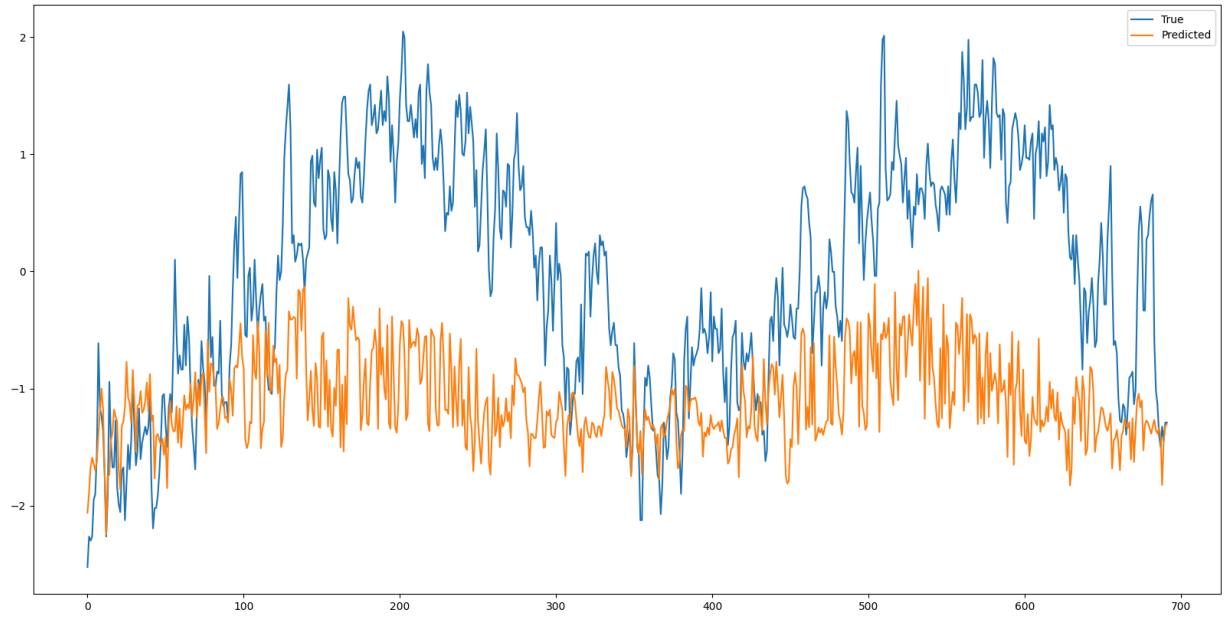
Таб. 2.7.1 Таблиця з відсортованими за кількістю помилок результатами проведення експерименту. MAE та RMSE по різному відсортували аттрибути 'date' та 'global_radiation'.

MAE	[('cloud_cover', 1.4076148), ('date', 1.2768903), ('global_radiation', 1.276356), ('pressure', 1.2504547), ('precipitation', 1.1395175), ('sunshine', 1.1358215), ('snow_depth', 1.1147891)]
RMSE	[('cloud_cover', 1.6761705), ('global_radiation', 1.5381536), ('date', 1.5318378), ('pressure', 1.5003419), ('precipitation', 1.3974767), ('sunshine', 1.3949977), ('snow_depth', 1.3647889)]

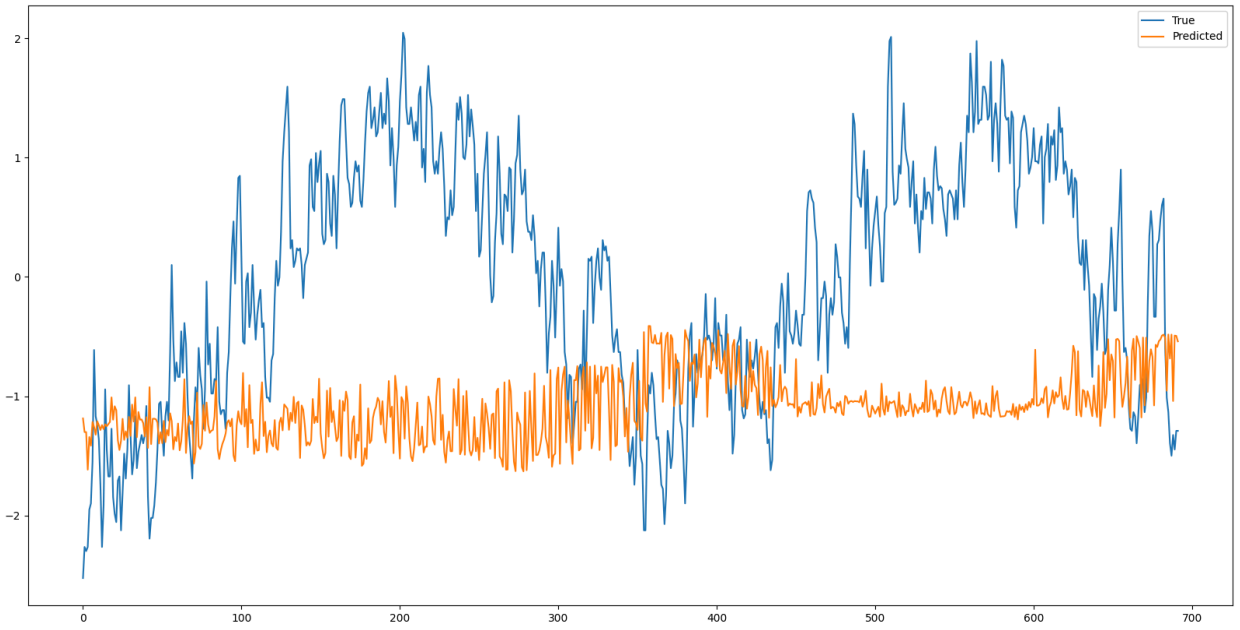
Таким чином ми отримали що хмарність, рівень радіації та дата найбільше зіпсувались при заміні на випадкові значення, а значить найбільше впливають на можливості передбачення. Відповідно ці параметри мають

найбільший вплив на цільовий атрибут. Сніг в свою чергу має найменший вплив.

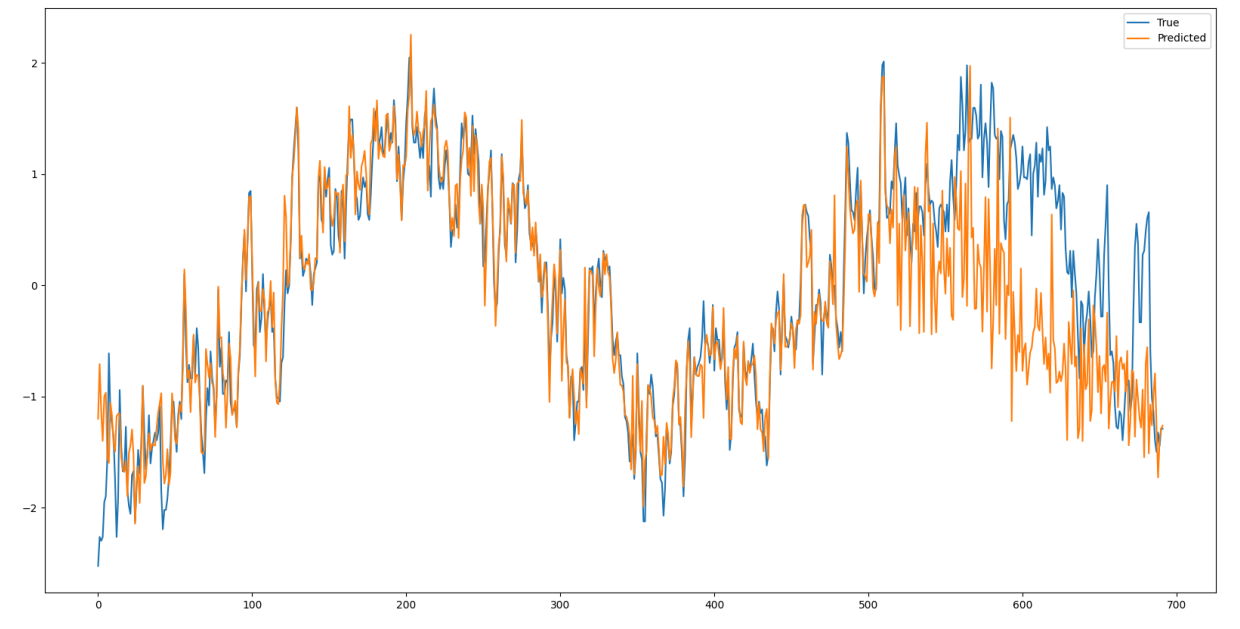
Далі приведені графіки того як змінюється передбачення при заміні на 0 дати, тиску, та снігу (синім позначено справжнє значення, помаранчевим - зпрогнозоване).



Мал. 2.7.1 - Підміна атрибуту “день спостереження” з вираженим погіршенням властивостей передбачення



Мал. 2.7.2 - Підміна атрибуту “атмосферний тиск” з вираженим погіршенням властивостей передбачення



Мал 2.7.3 - Підміна атрибуту “товщина снігу” з незначним впливом на якість передбачення моделі

Інші діаграми цього експерименту в Додатку 2.

На графіках і даних видно що заміна реальних даних на випадкову величину (0) для тих атрибутів які не мають вплив - графік не змінюється, для тих які мають вплив - графік спотворений. Гіпотеза підтверджена.

2.8 Експеримент 2.

Метою експеримента є перевірити на практичному прикладі можливість за допомогою результатів отриманих при роботі алгоритмов виводити нові гіпотези, тобто використовувати алгоритм як інструмент при розвідувального аналізу (Exploratory data analysis). Досліджуваний цільовий ряд у нас “Ціни на акції компанії твіттер”.

Крок 1. Підготовка даних

На етапі збору даних об’єднується 3 набори даних які мають теоретичний вплив на цільовий ряд.

1. TWITTER_2022.csv [10] - дані про ціни на акції компанії твітер за 2022 рік, з атрибутами Date, Open, High, Low, Close, Adj Close, Volume. З цих даних залишаються атрибут Close (ціна за акцію при закритті біржі) та Volume (кількість виторгуваних акцій за день роботи біржі).
2. ElonTweets_Sentiment.csv [11] - агригована статистика про твіти Ілона Маска. Атрибути: Datetime, Tweet Id, Text, Username, location, reply count, retweet count, like count, language, Twitter Access Point, Follower Count, Friends Count, verified, Date, mentions, sentiment (зі значеннями neutral, positive, negative). З цього набору даних збирається кількість коментарів з емоційним забарвленням positive, negative за день і зберігаю їх як нові атрибути positive та negative.
3. BigTech.csv [12] - ціни на акції схожих великих технологічних компаній, обрані за для того щоб візуалізувати вплив тенденцій ринку на утворення цін акцій твіттер. Атрибути Date, AMZN, META, NVDA, MSFT, GOOG, NFLX, AAPL.

Готова статистика має атрибути:

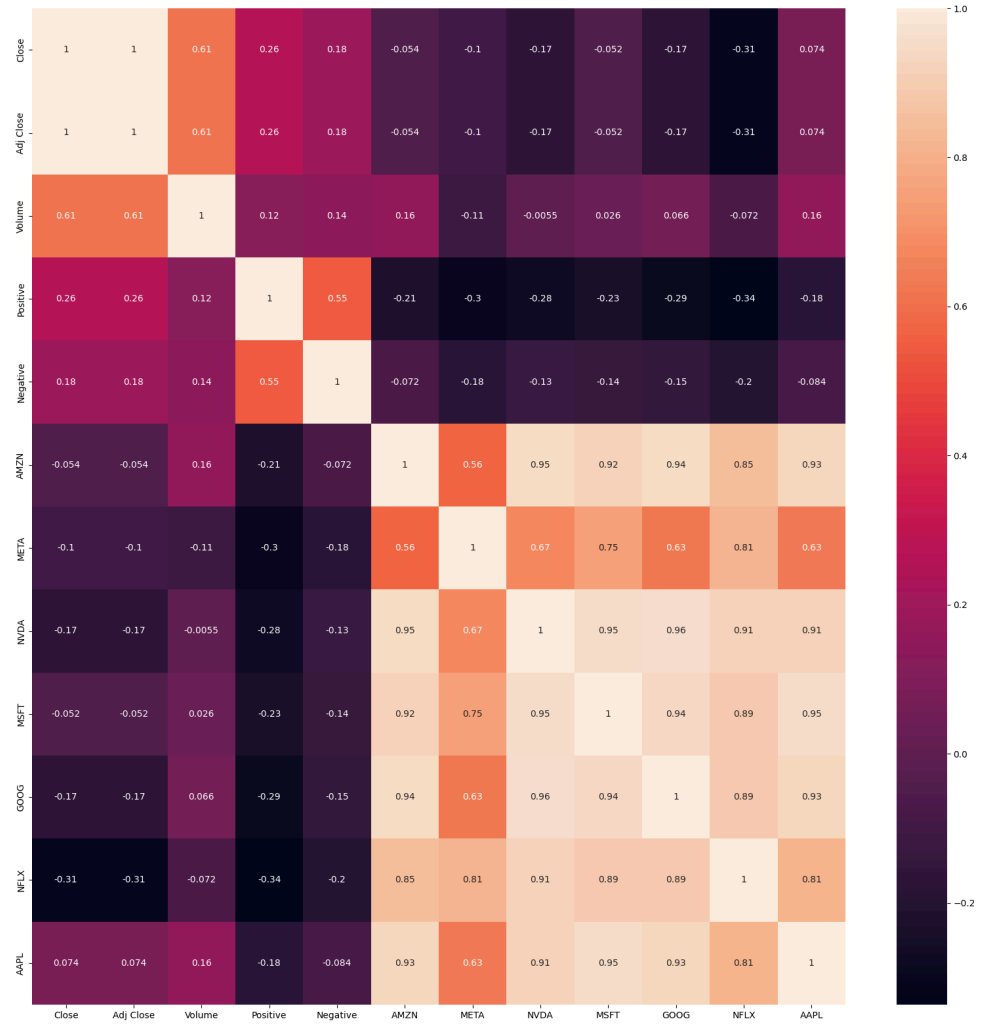
- Close - ціна за акцію при закритті біржі
- Volume - кількість виторгуваних акцій за день роботи біржі

- Positive - кількість коментарів від Ілона Маска з позитивним сентиментом.
- Negative - кількість коментарів від Ілона Маска з негативним сентиментом.
- AMZN - ціна за акцію Amazon
- META - ціна за акцію Meta
- NVDA - ціна за акцію NVidia
- MSFT - ціна за акцію Microsoft
- GOOG - ціна за акцію Alphabet
- NFLX - ціна за акцію Netflix
- AAPL - ціна за акцію Apple

Атрибути згруповані за параметром Date в діапазоні 2022 року, твіти Маска які припадають на вихідний день були закумульовані на перший робочий день біржі після вихідних.

Додаток 3. Приклад початкових даних з експеримента 2.

Матриця кореляції показує що технічні компанії крім META сильно корелюють між собою, відповідно при реальному застосуванні варто агрегувати або спростити їх.



Мал 2.8.1. Матриця кореляції властивостей з якої видно лінійну кореляцію між цінами на акції технічних компаній, крім META.

Крок 2. Тренування моделі нейронної мережі.

Нейронна модель побудована за допомогою компонентів пакету Tensorflow має сиквенційний вигляд і складається з шару 50 LSTM нейронів та одного вихідного Dense шару. Функція втрат - Середня абсолютна помилка (Mean Absolute Error, MAE). Метод оптимізації - ADAM. Модель тренувалась на тестових даних з кількість епох 200.

Модель була розділена на тренувальну і тестувальну добірку у співвідношенні 70 на 30. Окрім повної моделі також дослідження було проведено на підмножинах NVDA, MSFT, GOOG, NFLX, AAPL та Volume, Positive, Negative, AMZN, META.

Результати тренування моделі.

Повна множина:

MAE: 1.1453115940093994

RMSE: 1.3538013696670532

Train MAE: 1.1511173248291016

Train RMSE: 1.4148422479629517

Перша підмножина:

MAE: 0.284240186214447

RMSE: 0.3502415418624878

Train MAE: 1.0731295347213745

Train RMSE: 1.3923895359039307

Друга підмножина:

MAE: 0.604882538318634

RMSE: 0.7539247274398804

Train MAE: 1.1348215341567993

Train RMSE: 1.4716992378234863

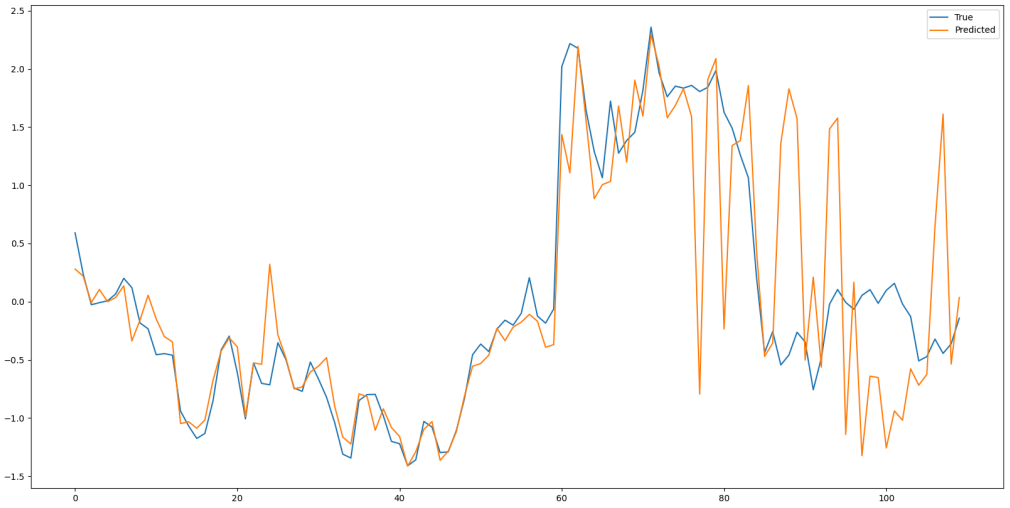
Крок 3. Оцінка

В даному прикладі випадковою величиною був мінімум атрибуту для якого відбувається заміна.

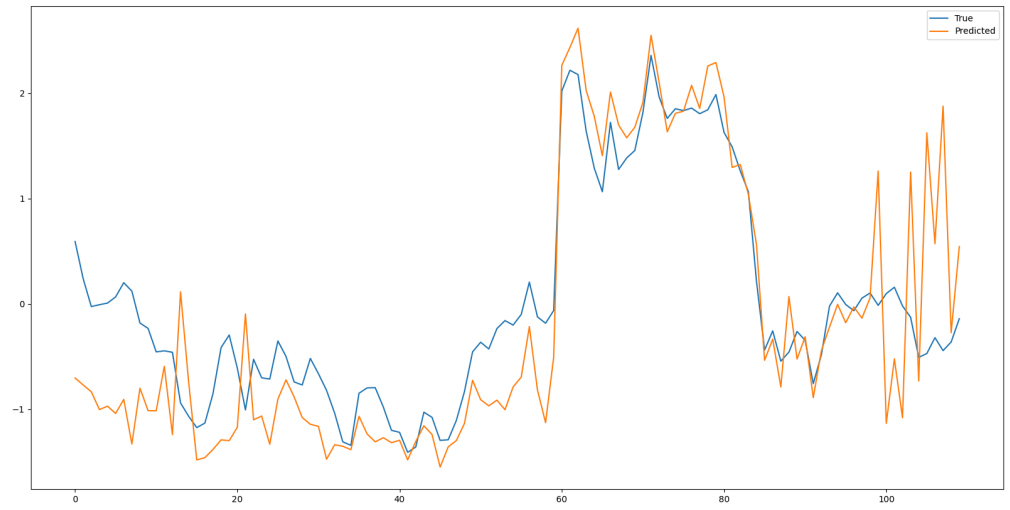
Результати відсортовані за кількістю отриманих помилок:

Таб. 2.8.1 Таблиця з відсортованими за кількістю помилок результатами проведення експерименту. MAE та RMSE по різному відсортували аттрибути 'Negative' та 'AMZN'.	
MAE	[('META', 1.1716533500000001), ('MSFT', 1.1437339), ('AMZN', 1.1019595500000001), ('Negative', 1.0941578), ('NVDA', 1.09261095), ('NFLX', 1.08611225), ('Positive', 1.05860965), ('AAPL', 1.02302197), ('GOOG', 1.0164838), ('Volume', 0.95554487)]
RMSE	[('META', 1.5107072000000001), ('MSFT', 1.4881921), ('Negative', 1.42154605), ('AMZN', 1.39309185), ('NVDA', 1.38640755), ('Positive', 1.3616104), ('NFLX', 1.3601376), ('AAPL', 1.3407540500000001), ('GOOG', 1.3369138), ('Volume', 1.2699810999999999)]

За результатом видно що найбільший вплив мали ціни акцій конкуруючого бізнесу 'META', серед акцій інших технологічних компаній система обрала 'MSFT’ хоча як ми пам’ятаємо з матриці кореляції це скоріше є результатом того що всі інші акції лінійно залежні і мережа на етапі тренування обрала 'MSFT' як найбільш зручніший з них. Початкова гіпотеза про негативні твіти Ілона маска опинилась аж на 4му місці, але і на це є виправдання якщо подивитись графіки сформовані підмножинами.



Мал. 2.8.2 Підміна атрибуту “Negative” з незначним впливом на якість передбачення моделі на початку, і значним впливом у другій половині графіку



Мал. 2.8.3 Підміна атрибуту Meta з незначним впливом на якість передбачення моделі посередині, і значним впливом на початку та вкінці

Інші діаграми цього експерименту в Додатку 4.

Увесь інтервал на якому в оригінальних даних (синій колір) має таку форму на ділянці підготовки продажа компанії, тому тут ціна була прив’язана до торгів. І графік негативних реакцій пана Маска відображає що його твіти не мали впливу на ціну акцій до моменту того як наближалось укладання угоди. В той самий час акції компанії Мета навпаки не мають впливу на діапазоні укладання угоди, зате впливають на моментах до та після що можна побачити через спотворення графіків на початку та вкінці.

Тож отримані в ході дослідження графіки можна використовувати як інструмент для виведення нових гіпотез.

2.9 Експеримент 3.

Стрестест. Прогнозування мавп’ячої віспи на середовищі утвореному міжнародними оцінками здоров’я в країнах світу.

У даному досліджені за мету ставиться дослідити поведінку алгоритму на наборі даних які дуже великі як для згорткових методів, так і для кореляції.

Крок 1. Підготовка даних

На етапі збору даних початкова добірка це статистика розповсюдження мавп’ячої віспи owid-monkeypox-data.csv [13] з атрибутами

Табл. 2.9.1 Перелік колонок в файлі owid-monkeypox-data			
location	iso_code	date	total_cases
total_deaths	new_cases	new_deaths	new_cases_smoothed
new_deaths_smoothed	new_cases_per_million	total_cases_per_million	new_cases_smoothed_per_million
new_deaths_per_million	total_deaths_per_million	new_deaths_smoothed_per_million	

З цієї добірки нашим цільовим атрибутом є new_cases_smoothed - усереднене на тиждень значення нових зареєстрованих хворих на мавп’ячу віспу. З цих даних ми використовуємо тільки new_cases_smoothed та location для подальшого агрегування даних з інших джерел.

Також з відкритих джерел було зібрано інформацію про щільність розселення людей у країні (density_kx100) та кількість населення в країні в тисячах (population_k).

Атрибутами середовища є оцінки велика тека даних World Health Statistics 2020 [14] від World Health Organization (WHO). Ця тека містить 39 файлів з статистикою охорони здоров'я для країн-членів ВООЗ. Ця статистика містить як кількісні величини проявів хвороб, смертності, очікуваних років життя так і відсоткові значення.

Загалом після нормалізації і очистки даних було отримано 100 атрибутів для даних з 63 країн. Матриця кореляцій розмірністю 100 на 100 показує що деякі поля які мали розділені по статтю на яку збирали статистику мають лінійну залежність 1 і в реальному застосуванні мають бути спрощені, але задля дослідження того як працює система залишимо їх.

Крок 2. Тренування моделі

Нейронна модель побудована за допомогою компонентів пакету Tensorflow має сиквенційний вигляд і складається з шару 50 LSTM нейронів та одного вихідного Dense шару. Функція втрат - Середня абсолютна помилка (Mean Absolute Error, MAE). Метод оптимізації - ADAM. Модель тренувалась на тестових даних з кількість епох 200.

Модель була розділена на тренувальну і тестувальну добірку у співвідношенні 60 на 40 країнах, тобто замість розбиття часового ряду кожної країни використовується часовий ряд одних країн для тренування, а інший для тестування. Таким чином маємо тестові країни Hungary, Serbia, Austria, Italy, Iceland, Slovakia, Costa Rica, France, Nigeria, Germany, Mexico, Portugal, Slovenia, China, Japan, Ghana та країни які ми застосовуємо для тренування Poland, Denmark, Finland, Malta, Panama, Dominican Republic, Ireland, Brazil, Croatia, Estonia, Chile, Greece, Norway, Peru, Romania, Latvia, Luxembourg, Jamaica, Canada, Barbados, Georgia, Argentina.

Також тренування здійснювалось як на повній множині атрибутів, так і на підмножинах. Розбивання підмножини відбувається таким чином що основна множина ділилась на підмножини розміром n.

```
deep = 20
    concrete_columns = []
    for i in range(0, len(regression_columns), deep):
        concrete_columns.append(regression_columns[i:i + deep])
```

Результати тренування для різних підмножин

Табл 2.9.2 Результати тренування моделі на повній кількості атрибутів, на підмножинах розміром 20, та на підмножинах розміром 10.	
Добірка	Оцінки
Повна	MAE: 0.36480209856570533 RMSE: 0.8731473299226797
розмір 20, ч.1	MAE: 0.33764255838163826 RMSE: 0.8654816206222662
розмір 20, ч.2	MAE: 0.36054242251614876 RMSE: 0.8728361450281983
розмір 20, ч.3	MAE: 0.3621760254745242 RMSE: 0.8755595742064094
розмір 20, ч.4	MAE: 0.33773967660486875 RMSE: 0.8630647514149754
розмір 20, ч.5	MAE: 0.33341421633519114 RMSE: 0.867587933044572
розмір 10, ч.1	MAE: 0.32165433300809027 RMSE: 0.8693606680844747
розмір 10, ч.2	MAE: 0.3307157823811715 RMSE: 0.8651642440851477
розмір 10, ч.3	MAE: 0.3411478939060558 RMSE: 0.8792735709303946
розмір 10, ч.4	MAE: 0.37054007681823714 RMSE: 0.8779033866399859
розмір 10, ч.5	MAE: 0.3614684857739235 RMSE: 0.8796820929003469
розмір 10, ч.6	MAE: 0.37417662576860405 RMSE: 0.8854168658530583
розмір 10, ч.7	MAE: 0.3255810824326943 RMSE: 0.8676270150621654
розмір 10, ч.8	MAE: 0.3664631829003045 RMSE: 0.8710603738573844

розмір 10, ч.9	MAE: 0.3383219806388215 RMSE: 0.8628011559545213
розмір 10, ч.10	MAE: 0.34099867803468287 RMSE: 0.8732196427826924

Варто зазначити що виконання навчання на такому великому наборі даних займає деякий великий час, і в цьому кореляція однозначно виграє.

Крок 3. Оцінка

В даному прикладі випадковою величиною було повністю випадкове число оскільки дані оцінок World Health Statistics для кожної з країн для якого відбувається заміна статичні, тому динамічне число буде ідеальним псуванням якості даних, хоча так само для отримання результату можна використати зсув значень на n.

Результати відсортовані за кількістю отриманих помилок:

Табл 2.9.3 Перші та останні атрибути відсортовані за помилками отримані в результаті підміна атрибуту випадковою величиною.

Назва атрибуту	MAE	RMSE
HALElifeExpectancyAtBirth|Period2019|Dim1Both sexes_First Tooltip	0.5983837	0.9504239625
basicDrinkingWaterServices|Period2016_First Tooltip	0.49567283249999994	0.9376537675
basicHandWashing|Period2016|Dim1Total_First Tooltip	0.44911031125	0.91048208

cleanFuelAndTech|Period 2018_First Tooltip	0.41008696875	0.9062209775
infantMortalityRate|Period2018|Dim1Male_First Tooltip	0.4089447025	0.9018697125
infantMortalityRate|Period2018|Dim1Female_First Tooltip	0.40416759375	0.8993018374999999
basicDrinkingWaterServices|Period2017_First Tooltip	0.40107058500000003	0.8880016125
infantMortalityRate|Period2016|Dim1Female_First Tooltip	0.31790908500000004	0.8644207175
under5MortalityRate|Period2018|Dim1Both sexes_First Tooltip	0.31541935875	0.8641162312499999
infantMortalityRate|Period2019|Dim1Female_First Tooltip	0.3153429925	0.86313282375
30-70cancerChd|Period2016|Dim1Male_First Tooltip	0.31094019500000003	0.86160184625

safelySanitization|Period_latest|Dim1Total_First Tooltip	0.2986793925	0.86152833
under5MortalityRate|Period2018|Dim1Male_First Tooltip	0.29477416624999997	0.8607805875000001

Алгоритм надав нам результати за якими ми зможемо значно зменшити нашу початкову множину атрибутів. Алгоритм працює значно повільніше за кореляцію, але при цьому алгоритм залишив рядки які корелюють між собою, але все ж необхідні для обчислення.

Мабуть все ж для такої великої множини атрибутів вигідніше застосовувати Principal Component Analysis (PCA) [15]. Алгоритм надто покладається на швидкодію нейронної мережі, а при наявності великого набору як даних так і атрибутів у нас зменшується якість тестування оскільки тренування йде з високою вірогідністю на шумах. В результаті досягнення якості, навіть з тими можливостями які надають LSTM шари, є важкодосяжним.

Правильніше в такому випадку використовувати алгоритм не як заміну фільтруючих методів, а навпаки користуватись ними в процесі підготовки даних.

ВИСНОВКИ

В ході проходження експериментів, була розроблена кодова база та алгоритм який за допомогою нейронних мереж може допомагати передбачати які з властивостей досліджуваного середовища впливають на досліджувану модель.

Мною було досліджено правдивість гіпотези про можливість визначення впливу властивостей середовища на часовий ряд за допомогою рекурентних нейронних мереж. Алгоритм може бути застосований для підтвердження гіпотез та для зменшення розмірності набору даних. Також алгоритм генерує достатню кількість інформації яку можна використовувати для формування нових гіпотез.

При перевірці його працездатності на великій кількості атрибутів бажаних результатів досягнуто не було, через що стверджується що алгоритм треба застосовувати не замість методів фільтрації, а разом з ними. Також за своїми властивостями алгоритм належить також до згорткових методів.

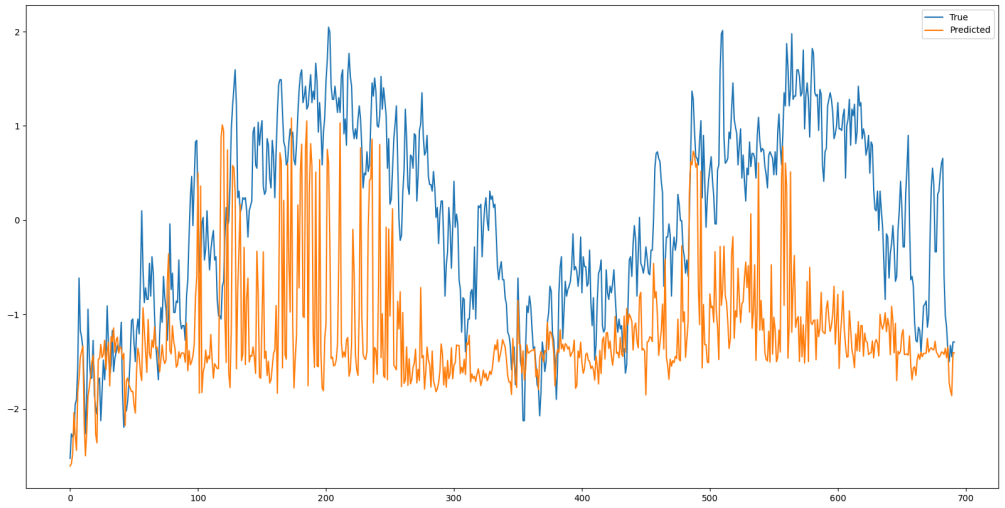
Було визначено актуальність проекту. Результати дослідження мають як теоретичну, так і практичну цінність. Робота може бути далі продовжена для кращої оптимізації розділення підмножин.

Система була реалізована з використанням Python та TensorFlow, так як ці системи мають очевидні переваги, згадані раніше.

ДОДАТОК 1. Приклад початкових даних з експеримента 1

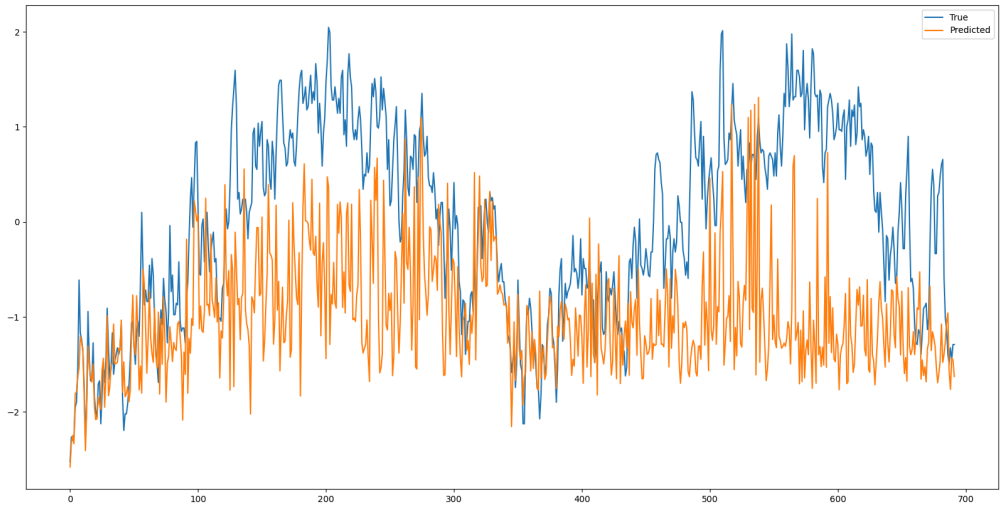
date	cloud_cover	sunshine	global_radiation	max_temp	mean_temp	min_temp	precipitation	pressure	snow_depth
19790101	2.0	7.0	52.0	2.3	-4.1	-7.5	0.4	101900.0	9.0
19790102	6.0	1.7	27.0	1.6	-2.6	-7.5	0.0	102530.0	8.0
19790103	5.0	0.0	13.0	1.3	-2.8	-7.2	0.0	102050.0	4.0
19790104	8.0	0.0	13.0	-0.3	-2.6	-6.5	0.0	100840.0	2.0
19790105	6.0	2.0	29.0	5.6	-0.8	-1.4	0.0	102250.0	1.0
19790106	5.0	3.8	39.0	8.3	-0.5	-6.6	0.7	102780.0	1.0
19790107	8.0	0.0	13.0	8.5	1.5	-5.3	5.2	102520.0	0.0
19790108	8.0	0.1	15.0	5.8	6.9	5.3	0.8	101870.0	0.0
19790109	4.0	5.8	50.0	5.2	3.7	1.6	7.2	101170.0	0.0
19790110	7.0	1.9	30.0	4.9	3.3	1.4	2.1	98700.0	0.0
19790111	1.0	6.8	55.0	2.9	2.6	0.3	2.3	98960.0	0.0
19790112	3.0	6.4	54.0	2.0	0.4	-2.0	0.0	100650.0	1.0
19790113	1.0	7.0	57.0	4.3	-2.6	-7.1	0.0	102350.0	1.0

ДОДАТОК 2. Графіки результатів експерименту 1, підміна атрибуту cloud_cover



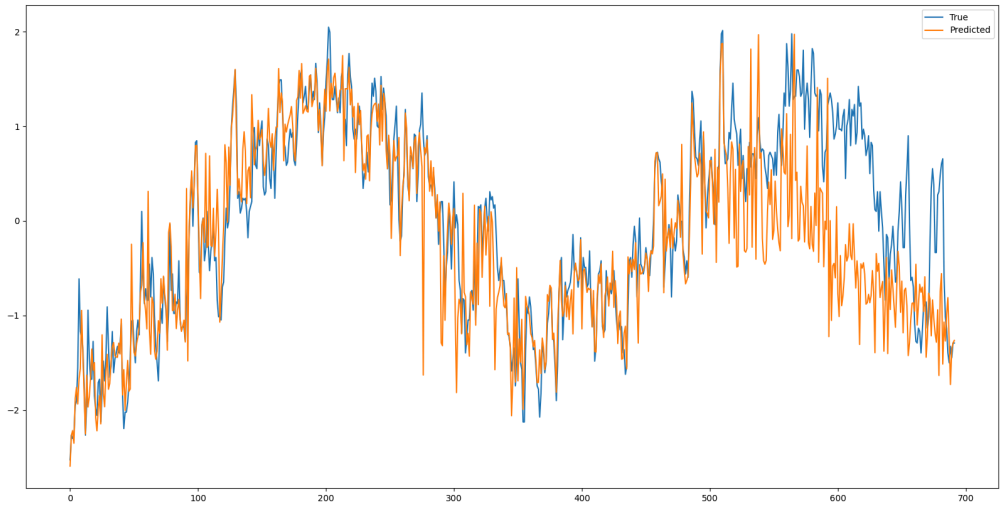
Підміна атрибуту cloud_cover

ДОДАТОК 3. Графіки результатів експерименту 1, підміна атрибуту global_radiation



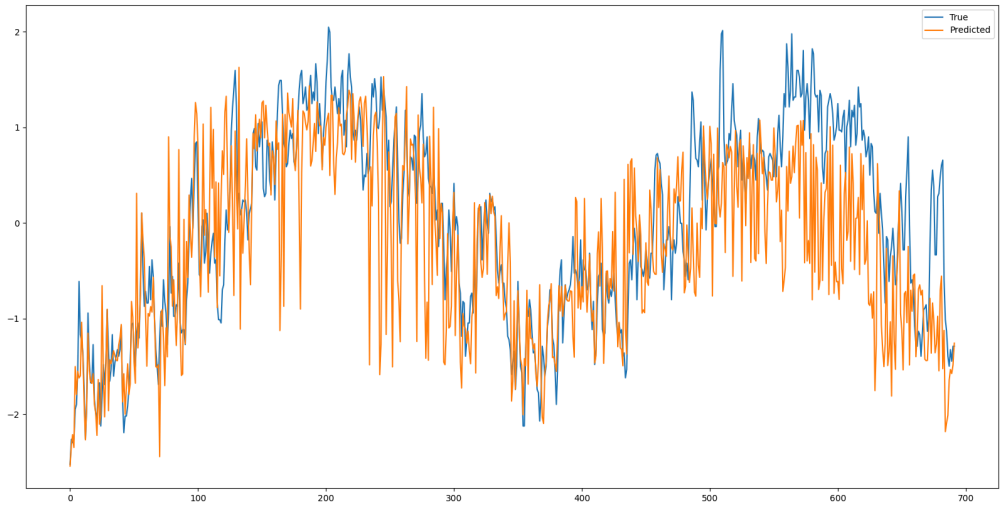
Підміна атрибуту global_radiation

ДОДАТОК 4. Графіки результатів експерименту 1, підміна атрибуту precipitation



Підміна атрибуту precipitation

ДОДАТОК 5. Графіки результатів експерименту 1, підміна атрибуту sunshine

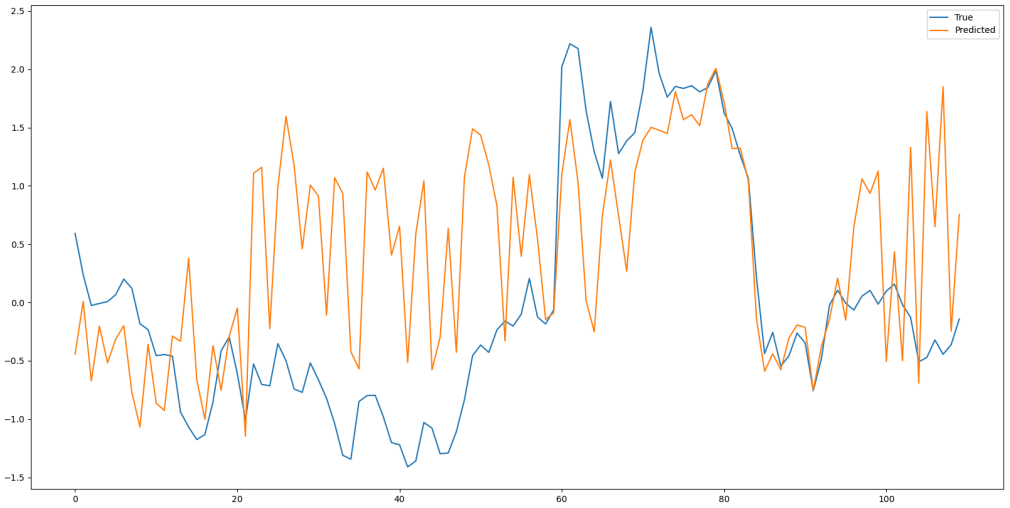


Підміна атрибуту sunshine

ДОДАТОК 6. Приклад початкових даних з експеримента 2.

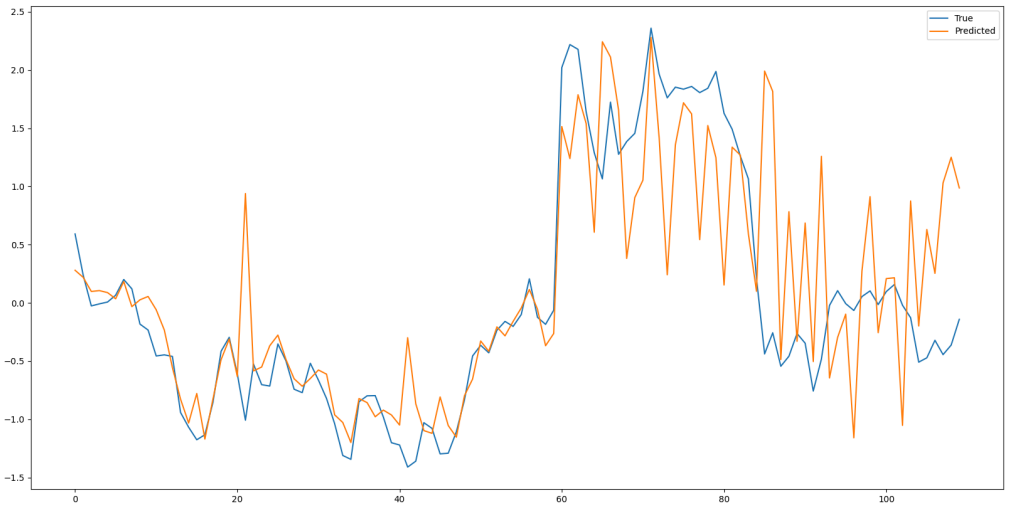
Close	Adj Close	Volume	Positive	Negative	AMZN	META	NVDA	MSFT	GOOG	NFLX	AAPL
42.66	42.66	144311900	0	0	170.404495	338.540009	301.209991	334.75	145.074493	597.369995	182.009995
40.849998	40.849998	214224400	0	0	167.522003	336.529999	292.899994	329.01001	144.416504	591.150024	179.699997
39.5	39.5	22008600	3	0	164.356995	324.170013	276.040009	316.380005	137.653503	567.52002	174.919998
39.59	39.59	16613400	0	0	163.253998	332.459991	281.779999	313.880005	137.550995	553.289978	172.0
39.669998	39.669998	14669900	1	1	162.554001	331.790009	272.470001	314.040009	137.004501	541.059998	172.169998
39.970001	39.970001	14997300	0	0	161.485992	328.070007	274.0	314.269989	138.574005	539.849976	172.190002
40.66	40.66	13815400	0	2	165.362	334.369995	278.170013	314.980011	140.017502	540.840027	175.080002
40.25	40.25	10439000	11	8	165.207001	333.26001	279.98999	318.269989	141.647995	537.219971	175.529999
38.700001	38.700001	15075600	4	0	161.214005	326.480011	265.75	304.799988	139.130997	519.200012	172.190002
38.439999	38.439999	14849800	0	0	162.138	331.899994	269.420013	310.200012	139.786499	525.690002	173.070007
37.299999	37.299999	14896800	2	1	158.917496	318.149994	259.029999	302.649994	136.290497	510.799988	169.800003

ДОДАТОК 7. Графіки результатів експерименту 2, графіки підмножин, підміна атрибуту AMZN



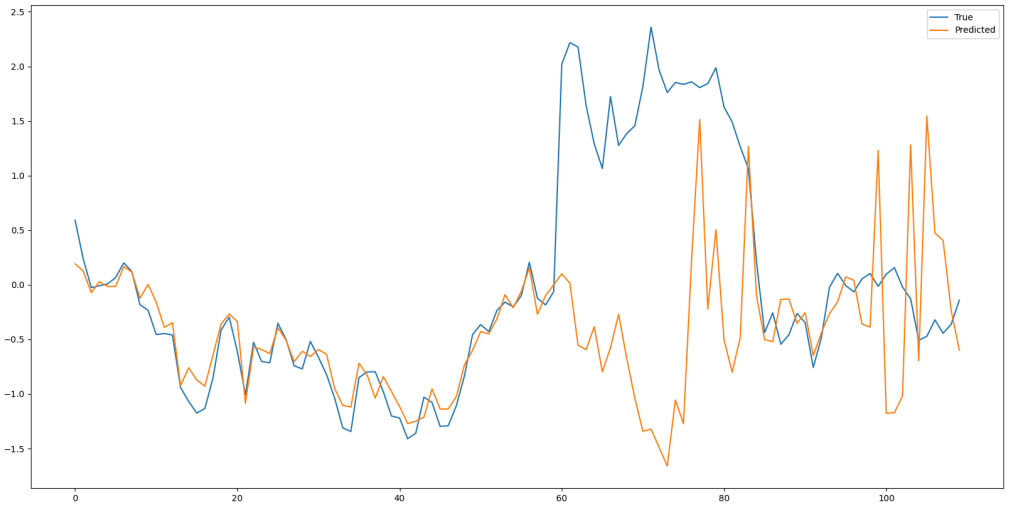
Підміна атрибуту AMZN

ДОДАТОК 8. Графіки результатів експерименту 2, графіки підмножин, підміна атрибуту Positive



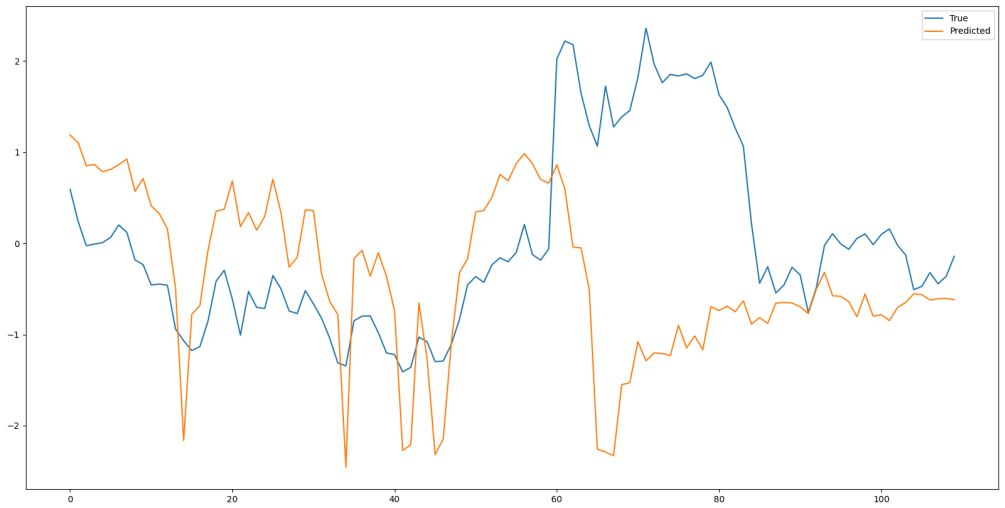
Підміна атрибуту Positive

ДОДАТОК 9. Графіки результатів експерименту 2, графіки підмножин, підміна атрибуту Volume



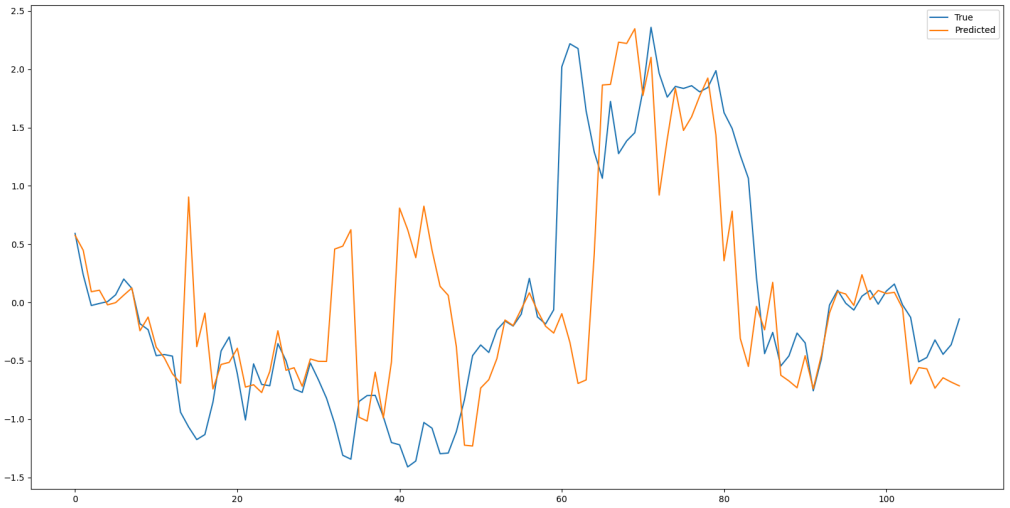
Підміна атрибуту Volume

ДОДАТОК 10. Графіки результатів експерименту 2, графіки підмножин, підміна атрибуту AAPL



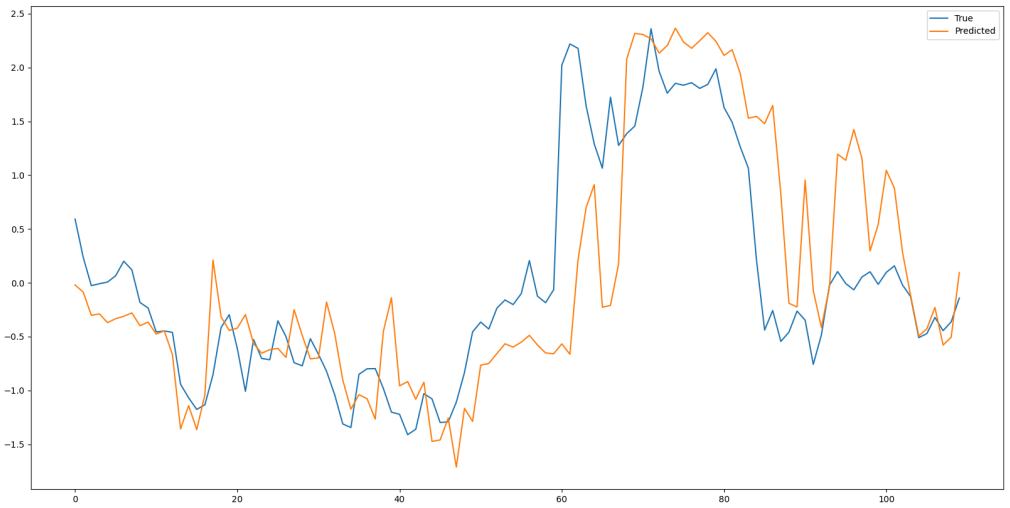
Підміна атрибуту AAPL

ДОДАТОК 11. Графіки результатів експерименту 2, графіки підмножин, підміна атрибуту GOOG



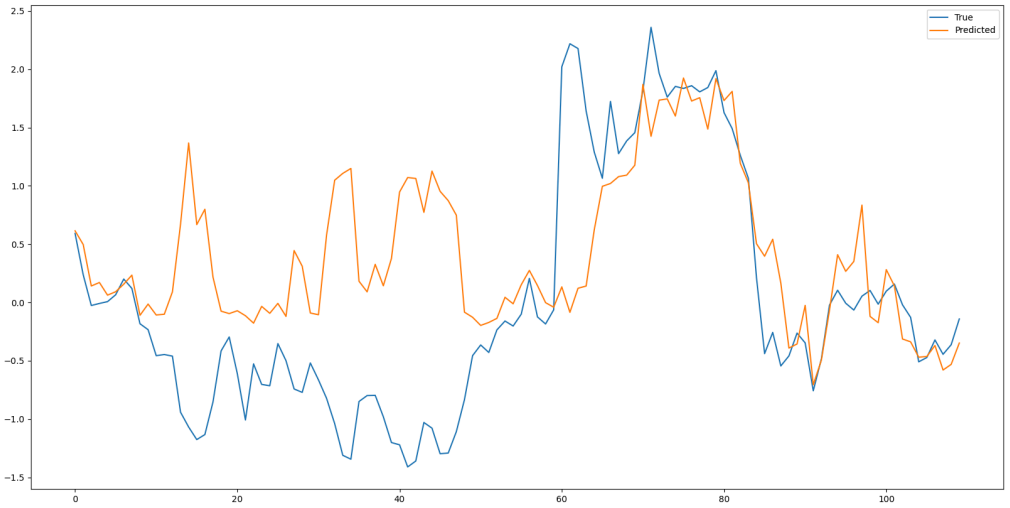
Підміна атрибуту GOOG

ДОДАТОК 12. Графіки результатів експерименту 2, графіки підмножин, підміна атрибуту MSFT



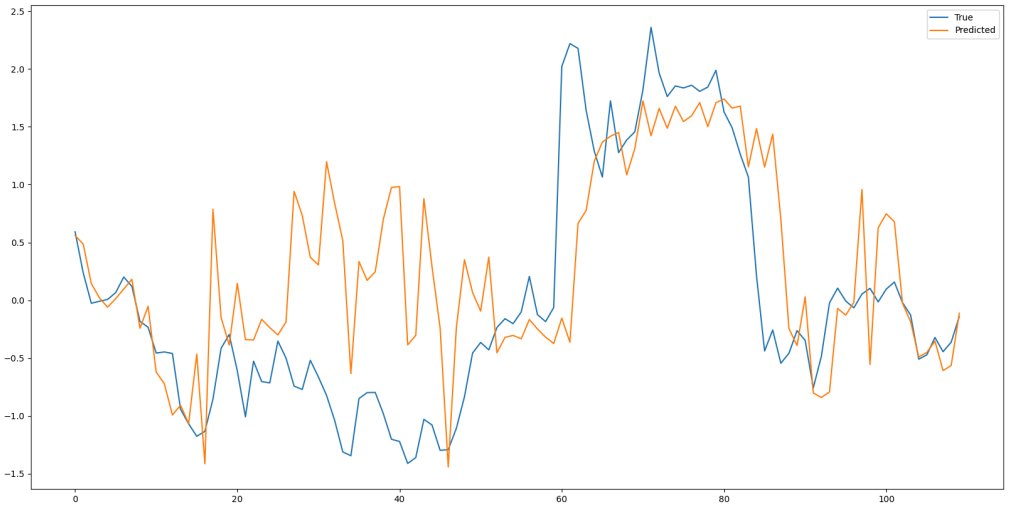
Підміна атрибуту MSFT

ДОДАТОК 13. Графіки результатів експерименту 2, графіки підмножин, підміна атрибуту NFLX



Підміна атрибуту NFLX

ДОДАТОК 14. Графіки результатів експерименту 2, графіки підмножин, підміна атрибуту NVDA



Підміна атрибуту NVDA

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. McKinsey & Company. (2022, December 6). *The state of AI in 2022—and a half decade in review*. McKinsey & Company. Retrieved May 13, 2024, from https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai-in-2022-and-a-half-decade-in-review
2. Davenport, T. H., & Bean, R. (2024, January 9). *Five Key Trends in AI and Data Science for 2024*. MIT Sloan Management Review. Retrieved May 13, 2024, from https://sloanreview.mit.edu/article/five-key-trends-in-ai-and-data-science-for-2024/
3. Keras. (n.d.). *LSTM layer*. Keras. Retrieved May 13, 2024, from https://keras.io/api/layers/recurrent_layers/lstm/
4. Qi, M., & Zhang, P. (2008, June). Trend Time–Series Modeling and Forecasting With Neural Networks. https://www.researchgate.net/publication/5384895_Trend_Time-Series_Modeling_and_Forecasting_With_Neural_Networks
5. Patarwal, P., Bala, R., & Bala, R. (2019, July). Financial and Non-Stationary Time Series Forecasting using LSTM Recurrent Neural Network for Short and Long Horizon. https://www.researchgate.net/publication/338361782_Financial_and_Non-Stationary_Time_Series_Forecasting_using_LSTM_Recurrent_Neural_Network_for_Short_and_Long_Horizon
6. Molnar, C. (2023, 08 21). *A Guide for Making Black Box Models Explainable*. 8.5 Permutation Feature Importance | Interpretable Machine Learning. Retrieved May 13, 2024, from https://christophm.github.io/interpretable-ml-book/feature-importance.html#feature-importance
7. Breiman, L. (2001). *“Random Forests.”Machine Learning 45 (1). Springer: 5-32 (2001)*.

8. *[1801.01489] All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously*. (2018, January 4). arXiv. Retrieved May 13, 2024, from http://arxiv.org/abs/1801.01489
9. MUKESH, K. (n.d.). *London_Weather_Data*. Wikipedia. Retrieved May 13, 2024, from https://www.kaggle.com/datasets/muku23/london-weather-data
10. MOTEFAKER, A. (n.d.). *Twitter Stock Market Dataset*. Kaggle. Retrieved May 13, 2024, from https://www.kaggle.com/datasets/amirmotefaker/twitter-stock-market-dataset
11. HUGGLER, A. (n.d.). *Elon Musk Tweets Sentiment(Classified via RoBERTa)*. Kaggle. Retrieved May 13, 2024, from https://www.kaggle.com/datasets/alexhuggler/elon-tweets-wsentimentclassified-via-roberta
12. MHASKE, A. (n.d.). *MAANG Stock Prices (July 2018 to July 2023)*. Kaggle. Retrieved May 13, 2024, from https://www.kaggle.com/datasets/adityamhaske/maang-stock-prices-july-2018-to-july-2023
13. Mathieu, E., Spooner, F., Dattani, S., Ritchie, H., & Roser, M. (n.d.). *Mpox*. Our World in Data. Retrieved May 13, 2024, from https://ourworldindata.org/monkeypox
14. ZEUS. (n.d.). *World Health Statistics 2020|Complete|Geo-Analysis*. Kaggle. Retrieved May 13, 2024, from https://www.kaggle.com/datasets/utkarshxy/who-worldhealth-statistics-2020-complete
15. Srivastava, A. (n.d.). *What Is Principal Component Analysis (PCA) & How It Works?* Analytics Vidhya. Retrieved May 13, 2024, from https://www.analyticsvidhya.com/blog/2016/03/pca-practical-guide-principal-component-analysis-python/