

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет водного господарства та природокористування
Навчально-науковий інститут автоматики, кібернетики та
обчислювальної техніки
Кафедра комп'ютерних технологій та економічної кібернетики

Допущено до захисту:

Завідувач кафедри

_____ д. е. н., проф. П. М. Грицюк

«_____» _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня «бакалавр»

за освітньо-професійною програмою «Інформаційні системи і технології»
спеціальності 126 «Інформаційні системи та технології»

на тему: «**Прогнозування динаміки фондового ринку з використанням
штучних нейронних мереж**»

Виконала:

здобувач вищої освіти 2 курсу із скороченим
терміном навчання, групи ІСТ-21інт

Сабанюк Ангеліна Юріївна

Керівник:

канд. техн. наук, доцент Гладка О. М.

Рецензент:

канд. техн. наук, доцент Барановський С. В.

Рівне – 2023

Національний університет водного господарства та природокористування
ННІ автоматики, кібернетики та обчислювальної техніки
Кафедра комп'ютерних технологій та економічної кібернетики

Освітньо-кваліфікаційний рівень – **бакалавр**

за освітньо-професійною програмою **«Інформаційні системи і технології»**

спеціальність **126 «Інформаційні системи та технології»**

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ д. е. н., проф. П. М. Грицюк

« _____ » _____ 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачу _____ Сабанюк Ангеліні Юріївні
(прізвище, ім'я, по-батькові)

1. Тема роботи _____ Прогнозування динаміки фондового ринку з
використанням штучних нейронних мереж

керівник роботи: _____ Гладка Олена Миколаївна, канд. техн. наук,
доцент
(прізвище, ім'я, по-батькові, науковий ступінь, вчене звання)

затверджена наказом по університету від “ 19 ” квітня 2023 р. С № 439

2. Термін здачі здобувачем закінченої роботи “ 09 ” червня 2023 р.

3. Вихідні дані до роботи розробити програмний застосунок, що реалізує
методи штучного інтелекту, та дослідити динаміку фондового ринку з
використанням штучної нейронної мережі ...

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх
належить розробити) вступ; аналіз предметної області; вибір та
обґрунтування засобів розробки програмного застосунку; розробка, опис і
дослідження створеної штучної нейронної
мережі.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових
креслень)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1.	Гладка О. М., доцент		
2.	Гладка О. М., доцент		
3.	Гладка О. М., доцент		

7. Дата видачі завдання “ 07 ” листопада 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Аналіз об’єкта дослідження, виявлення існуючих проблем	07.11.22 – 25.11.22	
2.	Аналіз існуючих інформаційних методів (технологій) вирішення проблеми	28.11.22 – 23.12.22	
3.	Вибір та обґрунтування засобів розробки програмного застосунку	30.01.23 – 24.02.23	
4.	Розробка, реалізація та дослідження штучної нейронної мережі	27.02.23 – 30.04.23	
5.	Підготовка тексту кваліфікаційної роботи	10.04.23 – 26.05.23	
6.	Підготовка презентації роботи	29.05.23 – 09.06.23	
7.	Відгук керівника, рецензування роботи, перевірка на плагіат	09.06.23 – 16.06.23	

Здобувач _____ **А. Ю. Сабанюк**
(підпис) (прізвище і ініціали)

Керівник кваліфікаційної роботи

(підпис)

(прізвище і ініціали)

О. М. Гладка

АНОТАЦІЯ

Сабанюк А. Ю Прогнозування динаміки фондового ринку з використанням штучних нейронних мереж . Кваліфікаційна робота на здобуття ступеня «бакалавр»: 50 с., 20рис., 2 додатки на 7 сторінок., 3 табл., 19 джерел.

Об'єктом дослідження є дані вартості цінних паперів компанії Amazon.

Предметом дослідження - є прогнозування динаміки фондового ринку з використанням штучних нейронних мереж

У процесі розробки були виконані наступні кроки: аналіз предметної області, побудова та аналіз нейронних мереж.

Для програмної реалізації було використано мову програмування Python, середовище розробки Jupyter Notebook.

Розроблений програмний продукт дозволяє виконувати аналіз штучних нейронних мереж з різними архітектурами, а також виконувати за їх допомогою прогнозування динаміки фондових показників компанії.

Ключові слова: штучна нейронна мережа, часовий ряд, Рекурентна нейронна мережа, Довготривала короткочасна пам'ять, Багатошаровий Персептрон, нейронна мережа, LSTM, RNN, MLP, прогнозування, Python, Jupyter Notebook.

ЗМІСТ

ВСТУП	7
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1. Прогнозування фондових показників.....	9
1.2. Використання штучних нейронних мереж для прогнозування фондового ринку	11
1.3. Часові ряди	13
1.4. Метрики для оцінки якості прогнозування.....	15
1.5. Алгоритми оптимізації.....	17
1.6. Функція активації	18
1.7. Опис вхідних даних для прогнозування.....	18
2. ОПИС НЕЙРОННИХ МЕРЕЖ ДЛЯ ПРОГНОЗУВАННЯ ФОНДОВИХ ПОКАЗНИКІВ	20
2.1. Штучні нейронні мережі.....	20
2.2. Рекурентні нейронні мережі.....	23
2.3. Довготривала короткочасна пам'ять	26
2.4. Багатошаровий Персептрон.....	26
3. РОЗРОБКА ТА РЕЗУЛЬТАТИ РЕАЛІЗАЦІЇ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ.....	29
3.1. Засоби та середовище розробки.....	29
3.2. Аналіз даних фондових показників компанії	33
3.3. Результати реалізації рекурентних нейронних мереж.....	36
3.4. Результати реалізації довгої короткочасної пам'яті.....	40
3.5. Результати реалізації Багатошарового Персептрона	43
ВИСНОВКИ.....	49
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	50
ДОДАТОК А.....	54
ДОДАТОК Б	57

ВСТУП

Прогнозування часових рядів є однією з ключових тем машинного навчання. Прогнозовані моделі на основі машинного навчання широко використовувалися в проектах часових рядів, необхідним для підприємств для полегшення прогнозування розподілу часу та ресурсів. Прогнозування часових рядів має багато застосувань, бізнес-планування, прогнози погоди та прогнозування курсу акцій.

Фондовий ринок є основою будь-якої економіки, основними цілями будь-яких інвестицій на фондовому ринку є максимізація прибутку та мінімізація ризику. Дуже важко передбачити ціни на акції через їх невизначену поведінку. Прогноз фондового ринку допомагає визначити рух фондових показників компанії та інших фінансових активів, якими торгують на біржі. Важко передбачити, як розвиватиметься фондовий ринок та знати, підніметься ціна акції вгору чи впаде. Інвестування на фондовому ринку може призвести до швидкого повернення інвестицій, тому прогнозування є однією із стратегій для отримання прибутку.

Багато різних факторів можуть впливати на ціни акцій, зокрема новини та результати діяльності компаній, показники галузі, настрої інвесторів та економічні фактори, через таку складність існує великий інтерес до застосування методів машинного навчання для оцінки великих історичних наборів даних фондових показників підприємств.

Метою роботи є розробка програмної реалізації та дослідження результатів навчань найбільш поширених штучних нейронних мереж.

Основними завданнями дипломної роботи аналіз предметної області, побудова нейронних мереж з різними архітектурами.

Об'єктом прогнозування є дані фондових показників компанії Amazon з сайту Yahoo! Finance за 5 років.

Предметом дослідження є прогнозування динаміки фондового ринку з використанням штучних нейронних мереж.

Кваліфікаційна робота складається з трьох розділів, у першому розділі

описано предметну область та підходи для прогнозування вартості цінних паперів. У другому розділі розглянуто штучні нейронні мережі за допомогою яких відбувається прогнозування фондових показників. В третьому розділі описано програмну реалізацію моделей штучних нейронних мереж з різними архітектурами.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Прогнозування фондових показників

Прогнозування ринку надає можливості отримання прибутку та є основним стимулом для більшості дослідників у цій галузі. Щоб передбачити ринок, більшість дослідників використовують технічний або фундаментальний аналіз.

Технічний аналіз зосереджується на аналізі напрямку цін, щоб передбачити майбутні ціни, аналізує вимірювані дані діяльності фондового ринку, такі як історичні прибутки та обсяг історичних торгів, тобто кількісну інформацію, яка могла б ідентифікувати торгові сигнали та вловити моделі руху фондового ринку. Технічний аналіз зосереджується на історичних і поточних даних, як і фундаментальний аналіз, але в основному він використовується для короткострокових торгових цілей [4].

Фундаментальний аналіз залежить від аналізу неструктурованої текстової інформації, наприклад фінансових новин і звітів про прибутки. Оцінює акції компанії шляхом вивчення її внутрішньої вартості, матеріальні активи, фінансову звітність, ефективність управління, стратегічні ініціативи та поведінку споживачів тобто всі основи компанії. Будучи відповідним показником для довгострокових інвестицій, фундаментальний аналіз спирається як на історичні, так і на теперішні дані для вимірювання доходів, активів, витрат, зобов'язань тощо [4].

На фондовому ринку існує два основних сегменти фундаментального аналізу:

Кількісний аналіз — це підхід до фундаментального аналізу акцій компанії, який зосереджується на використанні числових даних і статистичних методів для оцінки фінансових показників компанії. Він передбачає аналіз різних фінансових показників, таких як дохід, прибуток, грошовий потік і коефіцієнти, щоб визначити тенденції та моделі, які можуть допомогти інвесторам приймати обґрунтовані інвестиційні рішення.

Ключові методи кількісного аналізу, які зазвичай використовуються в аналізі запасів, включають:

- Аналіз співвідношення такі як співвідношення ціни та прибутку, рентабельність власного капіталу і співвідношення боргу до власного капіталу). Порівнюючи ці показники з галузевими показниками та історичними тенденціями, інвестори можуть отримати уявлення про прибутковість, ефективність компанії.

- Аналіз тенденцій використовується для аналізу різних фінансових показників, таких як дохід, прибуток і грошовий потік, протягом певного періоду часу, щоб оцінити потенціал зростання компанії та майбутні перспективи.

- Регресійний аналіз використовується для виявлення факторів, що впливають на фінансові показники компанії, таких як макроекономічні показники, галузеві тенденції та фактори конкуренції, і прогнозування її майбутніх фінансових показників на основі історичних даних.

- Аналіз дисконтованого грошового потоку це техніка, яка передбачає оцінку поточної вартості майбутніх грошових потоків компанії та може допомогти вам оцінити, переоцінені акції чи недооцінені.

- **Якісний аналіз** зосереджується на нефінансових, нематеріальних аспектах, які впливають на результати діяльності компанії. Він передбачає оцінку менеджменту компанії, корпоративної культури, галузевих тенденцій, конкурентної позиції, факторів ризику та інших пов'язаних якісних факторів, які можуть вплинути на довгострокові перспективи компанії. Якісний аналіз є важливим доповненням до кількісного аналізу, і разом вони дають цілісну оцінку компанії та її акцій.[3]

Прогнозування фондового ринку має на меті визначити майбутній рух вартості цінних паперів фінансової біржі. Точне передбачення руху ціни призведе до більшого прибутку, який інвестори можуть отримати. Передбачення того, як буде рухатися фондовий ринок, є одним із найскладніших питань на які впливає багато факторів, таких як процентні

ставки, політика, економічне зростання, які роблять фондовий ринок нестабільним і який дуже важко передбачити точно. Прогноз акцій пропонує величезні шанси на прибуток і є основною мотивацією для досліджень у цій галузі, знання руху акцій за частки секунди може призвести до високого прибутку [1]. Оскільки інвестиції в цінні папери є основною діяльністю фінансового ринку, відсутність точних знань і детальної інформації призведе до втрати інвестицій. Прогнозування фондового ринку є складним завданням, оскільки рух ринку завжди піддається невизначеності [2].[5]

1.2. Використання штучних нейронних мереж для прогнозування фондового ринку

Фондовий ринок привертає велику увагу інвесторів. Вловлення регулярності змін фондового ринку завжди було ключовим моментом для інвесторів та інвестиційних компаній. Велику зацікавленість інвесторів викликає сфера дослідження прогнозування курсу акцій. Щоб зробити успішне інвестування, багато інвесторів хочуть знати майбутнє фондового ринку.

Машинне навчання (МН) зосереджено на створенні комп'ютерних алгоритмів, Зокрема, алгоритми МН можуть розпізнавати закономірності та зв'язки між даними, з якими вони навчаються, будувати математичні моделі, що стосуються таких закономірностей, і використовувати ці моделі для прогнозування або прийняття рішень. Крім того, чим більше інформації зможуть обробити системи на основі МН, тим вищезгадані моделі стануть більш досконаліми, що дозволить алгоритмам покращити свою аналітичну та прогнозну продуктивність.

Такі можливості є важливими для фінансових компаній. Вивчаючи вхідні дані включно з тенденціями акцій, корпоративними показниками, фінансовими новинами, поведінкою інвесторів та інформацією з соціальних

мереж, ці моделі можуть точно визначити найтонші, нелінійні зв'язки між усіма цими змінними. На основі таких висновків вони створять реалістичні прогнози вартості цінних паперів та нададуть гравцям ринку корисну інформацію та рекомендації щодо майбутніх інвестицій. [6] Оцінка алгоритмів машинного навчання для прогнозів фондового ринку є завданням, до якого слід підходити з належною обережністю з двох вагомих причин. По-перше, дослідження все ще тривають і далекі від досягнення загальноновизнаних результатів, оскільки діапазон алгоритмів, придатних для цієї мети, досить широкий, і оцінка їхньої точності в різних сценаріях може бути досить складною. По-друге, корпорації та інвестиційні компанії зазвичай неохоче розкривають свої алгоритми машинного навчання, щоб зберегти конкурентну перевагу.

Тим не менш, можна отримати загальне уявлення про прогрес у розробці та впровадженні алгоритмів з академічних досліджень і звітів наукових товариств, зосереджених на різних методах прогнозування вартості цінних паперів:

- Машинне навчання охоплює такі алгоритми, як випадковий ліс, метод опорних векторів і K-найближчий сусід, а також аналіз часових рядів на основі машинного навчання за допомогою техніки авторегресійної інтегрованої моделі ковзного середнього.

- Глибоке навчання (Deep Learning) і нейронні мережі, включаючи рекурентні нейронні мережі (Recurrent neural networks, RNN), довгу короткочасну пам'ять (Long short-term memory, LSTM) і нейронні мережі графів (GNN).

Алгоритми машинного навчання показали відносно вищу точність, особливо при обробці широких наборів даних, і навіть більшу точність, коли об'єднані в гібридні моделі. Таке злиття різних алгоритмів машинного навчання може легко розширити їхній потенціал:

- Випадковий ліс (Random forest) - алгоритм, який забезпечує оптимальну точність для великих наборів даних і широко використовується в

прогнозуванні запасів для регресійного аналізу, а саме визначення зв'язків між кількома змінними.

- Метод опорних векторів (Support Vector Machines, SVM) - алгоритм, заснований на керуваному навчанні (навчання шляхом надання фактичних прикладів вхідних і вихідних даних), особливо точний із масивними наборами даних, але не такий хороший для обробки складних і динамічних сценаріїв.

- К-найближчий сусід (k nearest neighbor) - цей алгоритм використовує досить трудомістку техніку на основі відстані для прогнозування результату певної події на основі записів її найбільш подібних історичних ситуацій, які називаються «сусідами».

- Авторегресійна інтегрована модель ковзного середнього (AutoRegressive Integrated Moving Average, ARIMA) - техніка прогнозування часових рядів, яка найкраще працює з лінійними даними для прогнозування короткострокових коливань цін на акції на основі минулих тенденцій акцій, наприклад сезонність, але не може обробляти нелінійні дані та робити точні довгострокові прогнози.[6]

1.3. Часові ряди

Часовий ряд — це послідовність точок даних, які виникають у послідовному порядку протягом певного періоду часу. [12]

Прогнозування часових рядів – це техніка, яка використовує історичні та поточні дані для прогнозування майбутніх значень протягом певного періоду часу або певної точки в майбутньому. Проаналізувавши дані які були збережені в минулому, можна приймати обґрунтовані рішення.

Чимало галузей промисловості та наукових галузей використовують прогнозування часових рядів. Деякі з найбільш актуальних включають:

- бізнес планування;
- криптовалюти;
- фондові ринки;

- моделювання поширення хвороби;

- розпізнавання образів;
- розподіл ресурсів;
- спортивна аналітика;
- статистика;
- прогноз погоди. Існує чимало факторів, пов'язаних із прогнозуванням часових рядів, але найважливіші з них включають наступне:
 - Обсяг даних, є важливим фактором. Чим більше даних є тим краще модель генерує прогнози . Це також полегшує моделі розпізнавання тенденцій і шуму в даних.
 - Якість даних передбачає деякі базові вимоги, такі як відсутність дублікатів, стандартизований формат даних, а також те, щоб дані збиралися послідовно або через регулярні проміжки часу.
 - Сезонність означає, що існують чіткі періоди часу, коли дані містять постійні відхилення.
 - Тенденції – вони вказують, чи буде змінна в часовому ряді збільшуватися або зменшуватися в даний період. Також можливо розрахувати ймовірність тенденції, щоб приймати ще більш обґрунтовані рішення на основі даних.
 - Неочікувані події завжди можуть статися, це необхідно враховувати під час створення моделі прогнозування. Вони представляють шум в історичних даних, і вони також непередбачувані. [13]

Часовий ряд можна побудувати за допомогою будь-яких даних, які вимірюються протягом часу через рівномірні інтервали. Історичні курси акцій, прибутки, валовий внутрішній продукт або інші послідовності фінансових чи економічних даних.

1.4. Метрики для оцінки якості прогнозування

Оцінка точності є важливою частиною процесу створення моделей машинного навчання, щоб описати, наскільки добре модель виконує свої

прогнози. Після створення регресійної вона використовує частину вихідного набору даних для прогнозування результатів і порівняння їх із фактичними.

В кваліфікаційній роботі метриками для оцінки якості прогнозування було використано середню абсолютну похибку (MAE) та середньоквадратичну похибку (MSE):

- Середня абсолютна похибка (MAE) - середнє значення суми абсолютних різниць між прогнозованими та фактичними значеннями, загальний показник для оцінки того, наскільки точними є прогнози моделі.

Дана метрика вимірюється за формулою:

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i|$$

Де, Y_i – фактичне значення, \hat{Y}_i – прогнозоване значення, n – кількість спостережень.

Отримане значення являє собою середню величину помилок у передбаченнях моделі.

- Середня квадратична помилка (MSE) - це середнє значення суми квадратів різниці між прогнозованими та фактичними значеннями. Формула даної метрики має вигляд:

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

Де, Y_i – фактичне значення, \hat{Y}_i – прогнозоване значення, n – кількість спостережень.

Дана метрика застосовується у випадках, коли потрібно підкреслити великі помилки та вибрати модель, яка дає менше саме великих помилок. Великі значення помилок стають помітнішими за рахунок квадратичної залежності.

MAE та MSE є важливими показниками для оцінки ефективності моделі машинного навчання. Вимірюючи середню похибку в прогнозах моделі,

можна оцінити її точність і прийняти обґрунтовані рішення про те, як її покращити. Чим менше значення похибки тим точнішим є прогноз.

1.5. Алгоритми оптимізації

Оптимізатори націлені на підвищення ефективності навчання алгоритмів шляхом підвищення їх точності в задачах прогнозування. Вірний вибір оптимізатора є важливим для прогнозування нейронних мереж:

- Стохастичний градієнтний спуск (Stochastic Gradient Descent, SGD) використовує лише одну вибірку з усього набору даних для кожної ітерації. Під час кожної ітерації зразок вибирається після випадкового перемішування. Це дозволяє уникнути надлишкових обчислень у великому наборі даних для оновлення параметра, оскільки він обробляє одне оновлення за раз.

- RMSprop, середньоквадратичне розповсюдженням, є технікою оптимізації, заснованою на градієнтному спуску з імпульсом. Це полегшує роботу алгоритму в горизонтальному напрямку, роблячи більші кроки та підвищуючи швидкість навчання. Він обмежує вертикальні коливання та динамічно регулює швидкість навчання, вибираючи різні швидкості для різних параметрів.

- Adam, це алгоритм оптимізації, який розраховує швидкість адаптивного навчання окремих параметрів, використовуючи лише градієнти першого порядку. Він поєднує в собі можливості таких алгоритмів оптимізації, як RMSprop і AdaGrad. Добре працює з розрідженими градієнтами, функція, отримана від AdaGrad, і добре працює з мобільним та онлайн-середовищем, функція, отримана від RMSprop.

- Adamax, є різновидом методу оптимізації Adam. Він виявився стабільним у формі нескінченного порядку, що відрізняє його від інших оптимізаторів. Це адаптивна форма SGD і краща за SGD з точки зору нечутливості до вибору параметрів.[14]

Для прогнозування нейронних мереж було обрано метод оптимізації Adam.

1.6. Функція активації

Функції активації в нейронних мережах і глибоке навчання відіграють важливу роль у запаленні прихованих вузлів для отримання більш бажаного результату. Основна мета функції активації — ввести в модель властивість нелінійності.[15] Для навчання нейронних мереж була використана функція активації ReLU (Rectified linear unit).

Виправлена лінійна одиниця (ReLU) — це функція активації, яка вводить властивість нелінійності в модель глибокого навчання та вирішує проблему зникаючих градієнтів. Функція ReLU проста і не вимагає важких обчислень (рис 1.1).

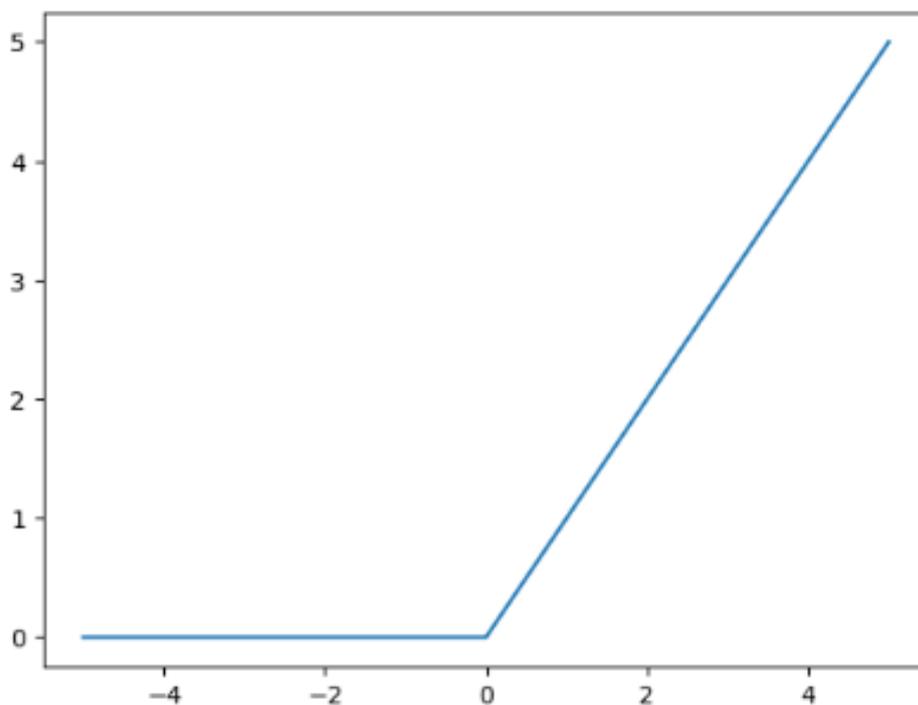


Рис 1.1 Функція ReLU

1.7. Опис вхідних даних для прогнозування

В дипломній роботі використано ціни на закриття акцій компанії Amazon (AMZN) за останні 5 років (15-05-2018 до 12-05-2023). Дані аналізу завантажувалися з сайту Yahoo!Finance, який пропонує історичні данні і дані фондового ринку в реальному часі.

Файл містить дані для:

- Date: Показує дату у форматі: рр-мм-дд
- Open: відображає ціну акції на відкритті фондового ринку,
- High: Показує найвищу ціну, досягнуту в цей день,
- Low: Показує найнижчу ціну, досягнуту в цей день,
- Close, Adj Close: відображає ціну акції на закриття фондового ринку,,
- Volume: відображає кількість акцій, якими торгували в цей день

Дані індексуються за датою, кожен рядок представляє дані за різну дату (рис 1.2).

Date	Open	High	Low	Close	Adj Close	Volume
2018-05-15	79.389999	79.389999	78.261002	78.806000	78.806000	101550000
2018-05-16	78.875000	79.721497	78.833504	79.363998	79.363998	51412000
2018-05-17	79.028000	79.702003	78.650002	79.087997	79.087997	42952000
2018-05-18	79.066498	79.179497	78.605003	78.718498	78.718498	52852000
2018-05-21	79.250000	79.602501	78.750000	79.273003	79.273003	58504000

Рис 1.2. Вхідні дані для прогнозування

На графіку нижче зображено максимум і мінімум акцій Amazon за 5 років (рис 1.3).

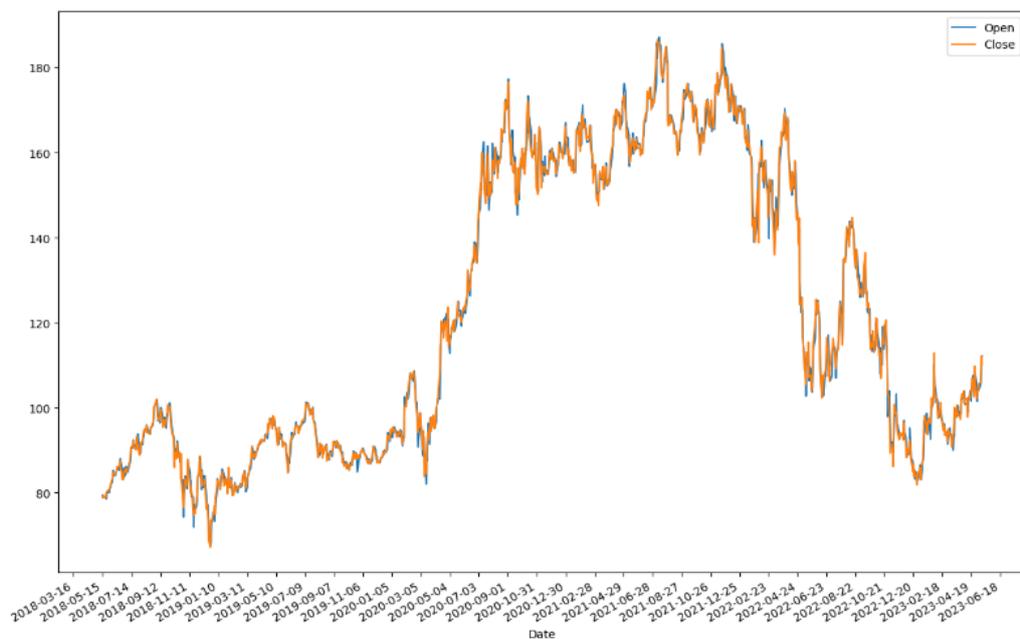


Рис 1.3 Максимум і мінімум вартості фондових показників

2. ОПИС НЕЙРОННИХ МЕРЕЖ ДЛЯ ПРОГНОЗУВАННЯ ФОНДОВИХ ПОКАЗНИКІВ

2.1. Штучні нейронні мережі

Штучна нейронна мережа — це система обробки інформації, яка має певні характеристики продуктивності, спільні з біологічними нейронними мережами. Штучні нейронні мережі були розроблені як узагальнення математичних моделей людського пізнання або нейронної біології, на основі таких припущень:

- Обробка інформації відбувається в багатьох простих елементах, які називаються нейронами.

- Сигнали передаються між нейронами через з'єднувальні зв'язки.

- Кожна ланка з'єднання має пов'язану вагу, яка примножує переданий сигнал.

- Кожен нейрон застосовує функцію активації (зазвичай нелінійну) до свого чистого входу (сума зважених вхідних сигналів), щоб визначити його вихідний сигнал.

Нейронна мережа характеризується:

- шаблоном зв'язків між нейронами (архітектурою);

- методом визначення вагових коефіцієнтів зв'язків (алгоритм навчання)

- функцією активації.

Нейронна мережа складається з великої кількості простих елементів обробки: нейронами або вузлами. Кожен нейрон з'єднаний з іншими нейронами за допомогою спрямованих комунікаційних зв'язків, кожен з яких має відповідну вагу. Ваги представляють інформацію, яка використовується мережею для вирішення проблеми

Кожен нейрон має внутрішній стан, який називається його активацією або рівнем активності, який є функцією вхідних даних, які він отримав. Як правило, нейрон посилає свою активацію як сигнал до кількох інших нейронів. Нейрон може посилати лише один сигнал за раз, хоча цей сигнал транслюється до кількох інших нейронів.

Штучний нейрон або нейронний вузол є математичною моделлю. У більшості випадків він обчислює середньозважену вхідну інформацію, а потім застосовує до неї зміщення. Після цього він пропускає результуючий термін через функцію активації. Ця функція активації є нелінійною функцією, такою як сигмоїдна функція, яка приймає лінійний вхід і дає нелінійний вихід.

Типова нейронна мережа складається з шарів нейронів, які називаються нейронними вузлами. Ці шари бувають трьох типів:

- вхідний шар (один) - це перший рівень типової нейронної мережі. Нейрони вхідного рівня отримують вхідну інформацію, обробляють її за допомогою математичної функції (функції активації) і передають вихідні дані нейронам наступного рівня на основі порівняння з попередньо встановленим пороговим значенням. Зазвичай у мережі є лише один вхідний рівень. Попередньо обробляється текст, зображення, аудіо, відео та інші типи даних, щоб отримати їх числове представлення. Пізніше передається це числове представлення як інформацію кожному нейрону вхідного рівня. Потім кожен нейрон застосовує попередньо визначену нелінійну математичну функцію для обчислення результату. На останньому етапі масштабується результат за попередньо встановленими ваговими коефіцієнтами, пов'язаними з кінцями між нейронами вихідного шару та відповідними нейронами вхідного шару.

- прихований шар – у нейронній мережі може бути один або декілька прихованих шарів. Нейрони в прихованому шарі отримують дані або від нейронів вхідного шару, або від нейронів попередньо прихованого шару. Потім кожен нейрон передає вхідні дані іншій нелінійній функції активації, а після цього надсилає вихідні дані нейронам наступного рівня.

- вихідний шар - у мережі є лише один вихідний рівень, який визначає завершення нейронної мережі. Подібно до описаних раніше шарів, нейрони у вихідному шарі отримують вхідні дані від попередніх шарів. Потім вони

обробляють їх за допомогою нових функцій активації та виводять очікувані результати. Залежно від типу штучної нейронної мережі, можна або

використовувати цей вихід як кінцевий результат, або подати його як вхід до тієї або іншої нейронної мережі.

Кожен нейронний вузол пов'язаний з іншим і характеризується своєю вагою та порогом. Він отримує вхідні дані, на яких виконує певне перетворення, після чого надсилає вихідні дані. Якщо вихід будь-якого окремого вузла перевищує вказане порогове значення, цей вузол активується. Потім він надсилає дані на наступний рівень мережі. В іншому випадку він залишається неактивним і, отже, не передає жодних даних на наступний рівень мережі.[7]

2.2. Рекурентні нейронні мережі

Повторювана нейронна мережа (Recurrent neural networks , RNN) — це тип штучної нейронної мережі, яка використовує послідовні дані або дані часових рядів. Ці алгоритми глибокого навчання зазвичай використовуються для порядкових або часових проблем, таких як переклад, розпізнавання та обробка мови, підписи до зображень. Вони відрізняються своєю пам'ятю, оскільки вони беруть інформацію з попередніх вхідних даних, щоб впливати на поточні вхідні та вихідні дані. В традиційні глибоких нейронних мережах входи та виходи незалежні один від одного, вихід рекурентних нейронних мереж залежить від попередніх елементів у послідовності.

Іншою відмінною характеристикою рекурентних мереж є те, що вони спільно використовують параметри на кожному рівні мережі. У той час як мережі прямого зв'язку мають різні ваги для кожного вузла, рекурентні нейронні мережі мають однаковий параметр ваги на кожному рівні мережі. Тим не менш, ці ваги все ще коригуються за допомогою процесів зворотного поширення та градієнтного спуску, щоб полегшити навчання з підкріпленням.

Повторювані нейронні мережі використовують алгоритм зворотного поширення в часі для визначення градієнтів, який дещо відрізняється від

традиційного зворотного поширення, оскільки він специфічний для даних послідовності. Принципи алгоритму зворотного поширення в часі такі ж, як і

традиційного зворотного поширення, коли модель навчається, обчислюючи помилки від вихідного до вхідного рівня. Ці розрахунки дозволяють правильно налаштувати та підігнати параметри моделі.

Алгоритм зворотного поширення в часі відрізняється від традиційного підходу тим, що підсумовує помилки на кожному кроці часу, тоді як в інших мережах не потрібно підсумовувати помилки, оскільки вони не поділяють параметри на кожному рівні. Через цей процес в RNN, виникає дві проблеми, такі як вибухові градієнти та зникнення градієнтів. Ці проблеми визначаються розміром градієнта, який є нахилом функції втрат уздовж кривої помилок. Коли градієнт занадто малий, він продовжує зменшуватися, оновлюючи вагові параметри, поки вони не стають незначними, тобто 0. Коли це відбувається, алгоритм більше не навчається.

Вибухові градієнти виникають, коли градієнт занадто великий, що створює нестабільну модель. У цьому випадку ваги моделі виростуть занадто великими, і в кінцевому підсумку вони будуть представлені як NaN. Одним із рішень цих проблем є зменшення кількості прихованих шарів у нейронній мережі, усунення частини складності в моделі RNN.[]

Існують різні архітектури RNN, які застосовуються в задачах машинного навчання:

- Двонаправлені рекурентні нейронні мережі (BRNN)

У BRNN вхідні дані з майбутніх часових кроків використовуються для підвищення точності мережі.

- Gated Recurrent Units (GRU)

Ці мережі розроблені для вирішення проблеми зникнення градієнта. Вони мають шлюз скидання та оновлення та визначають, яку інформацію слід зберігати для майбутніх прогнозів.

- Довгокороткочасна пам'ять

Були розроблені для вирішення проблеми зникнення градієнта в RNN. [8]

2.3. Довготривала короткочасна пам'ять

Модель довготривалої короткочасної пам'яті (Long short-term memory, LSTM) є підтипом рекурентних нейронних мереж (RNN). Використовується для розпізнавання шаблонів у послідовностях даних, що з'являються в даних датчиків, курсах акцій або природній мові.

Проблема з рекурентними нейронними мережами полягає в тому, що вони мають короткочасну пам'ять для збереження попередньої інформації в поточному нейроні. Однак ця здатність дуже швидко зменшується для довших послідовностей. Щоб вирішити цю проблему, були представлені моделі LSTM, які дозволяють зберігати минулу інформацію ще довше.

Повторювані нейронні мережі і зберігають попередні дані у своїй короткочасній пам'яті. Коли пам'ять у ньому вичерпується, вони просто видаляють найдовше збережену інформацію та замінюють її новими даними. Модель LSTM намагається уникнути цієї проблеми, зберігаючи вибрану інформацію в довготривалій пам'яті. Ця довготривала пам'ять зберігається в так званому стані клітини. Крім того, існує також прихований стан, в якому зберігається короткочасна інформація з попередніх кроків обчислення. Прихований стан - це короткочасна пам'ять моделі.

На кожному кроці обчислення використовується поточний вхід, попередній стан короткочасної пам'яті і попередній стан прихованого стану.[18]

2.4. Багатошаровий Персептрон

Штучні нейронні мережі (ШНМ) спочатку були розроблені як математичні моделі можливостей біологічного мозку з обробки інформації. Основною структурою ШНМ є мережа невеликих блоків обробки, або вузли, які з'єднані один з одним зваженими зв'язками. З точки зору біологічної моделі, вузли представляють нейрони, а ваги зв'язків представляють силу синапсів між нейронами.

Мережа активується шляхом надання вхідних даних деяким або всім вузлам, і ця активація потім поширюється по всій мережі вздовж зважених з'єднань. Одна важлива відмінність між ШНМ, з'єднання яких утворюють цикли, і тими, чиї з'єднання є ациклічними. ШНМ із циклами називають рекурсивними або рекурентними нейронними мережами зворотного зв'язку, ШНМ без циклів називаються нейронними мережами прямого зв'язку. Добре відомі приклади нейронних мереж прямого зв'язку включають перцептрони, мережі радіальних базисних функцій, карти Кохонена і мережі Хопфілда. Найбільш широко використовуваною формою нейронних мереж прямого зв'язку, є багатошаровий перцептрон (Multi-Layer Perceptron, MLP).

Блоки в багатошаровому перцептроні розташовані в шари, до з'єднання, що просуваються з одного шару на наступний. Вхідні шаблони представлені на вхідному рівні, а результуючі активації блоків поширюються через приховані шари на вихідний рівень. Цей процес відомий як прямий перехід мережі. Блоки в прихованих шарах мають зазвичай нелінійні функції активації. Оскільки поведінка функції параметризована вагами з'єднання, один MLP здатний створювати екземпляри багатьох різних функцій. MLP з одним прихованим шаром, що містить достатню кількість нелінійних одиниць, може апроксимувати будь-яку безперервну функцію на компактній вхідній області з довільною точністю. [19]

Багатошаровий перцептрон доповнення до нейронної мережі прямого зв'язку. Він має три шари: вхідний, прихований і вихідний. Вхідний рівень приймає сигнал для обробки. Вихідний рівень відповідає за такі функції, як класифікація та прогнозування. Безпосередньо зв'язаний механізм MLP - це нескінченний ряд прихованих шарів, розташованих між вихідним і вхідним шарами. Дані проходять по прямому шляху від вхідного до вихідного рівня в MLP, еквівалентно мережі з прямою передачею. Техніка навчання зворотного поширення використовується для навчання всіх вузлів у MLP. MLP можуть вирішувати проблеми, які не є лінійно роздільними та структуровані відповідно до наближення кожної безперервної функції. [19]

Навчання ШНМ— це процес налаштування параметрів мережі, які відіграють ключову роль на етапі прогнозування в ШНМ. Таким чином, процес навчання описується наступним чином: після обчислення виходу нейронною мережею значення передається на вихід і обчислюється помилка . Тобто, ці процеси повторюються таким чином, щоб отриманий результат і бажаний були якомога ближчими, а потім приймаються . MLP і RNN застосовують однаковий процес навчання алгоритму, винятком є те, що у випадку RNN прихований вміст має можливість оброблятися за таким же чином, як і приховані і вихідні шари.

3. РОЗРОБКА ТА РЕЗУЛЬТАТИ РЕАЛІЗАЦІЇ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

3.1. Засоби та середовище розробки

Для побудови та навчання штучних нейронних мереж було обрано мову програмування Python та середовище розробки Jupyter Notebook.

Python пропонує стислий і читабельний код. Хоча за машинним навчанням і штучним інтелектом стоять складні алгоритми та різноманітні робочі процеси, простота Python дозволяє розробникам створювати надійні системи. Замість того, щоб зосереджуватися на технічних нюансах мови, розробники докладають усіх зусиль для вирішення проблеми машинного навчання.

Крім того, Python привабливий для багатьох розробників, оскільки його легко вивчити. Код Python зрозумілий людям, що полегшує створення моделей для машинного навчання.

Python більш інтуїтивно зрозумілий, ніж інші мови програмування. Багато фреймворків, бібліотек і розширень, які спрощують реалізацію різних функцій. Оскільки Python є мовою загального призначення, він може виконувати низку складних завдань машинного навчання та дає змогу швидко створювати прототипи, які дозволяють тестувати продукт для цілей машинного навчання.

Впровадження алгоритмів машинного навчання може бути складним і вимагає багато часу. Щоб скоротити час розробки, розробники звертаються до ряду фреймворків і бібліотек Python.

Бібліотека програмного забезпечення — це попередньо написаний код, який розробники використовують для вирішення типових завдань програмування. Python зі своїм багатим набором технологій має великий набір бібліотек для штучного інтелекту та машинного навчання: [17]

- *TensorFlow* фреймворк машинного та глибокого навчання. Це безкоштовна бібліотека з відкритим кодом, повністю заснована на мові

програмування Python і використовується для числових обчислень і потоку даних, що робить машинне навчання швидшим і легшим. TensorFlow може навчати та запускати глибокі нейронні мережі для розпізнавання зображень, класифікації рукописних цифр, рекурентної нейронної мережі, вбудовування слів, обробки природної мови, виявлення відео та багато іншого. TensorFlow складається з двох слів, Tensor і Flow:

- Тензор — це багатовимірний масив
- Потік - використовується для визначення потоку даних під час операції.

TensorFlow використовується для визначення потоку даних під час роботи з багатовимірним масивом або Tensor. Назва TensorFlow походить від його основного фреймворку Tensor. Тензор — це вектор або n-вимірна матриця, яка представляє всі типи даних. Усі значення в тензорі мають подібний тип даних із відомою формою. Форма даних — це розмірність матриці або масиву. Тензор може бути згенерований із вхідних даних або результату обчислення. У TensorFlow усі операції виконуються всередині графа. Група - це набір обчислень, що відбувається послідовно. Кожна транзакція називається підключеним вузлом операцій. TensorFlow використовує структуру графів.

Його було виправлено для запуску на кількох центральних або графічних процесорах і мобільних операційних системах. Переносимість графа дозволяє зберегти обчислення для поточного чи подальшого використання. Графік можна зберегти, оскільки його можна буде виконати в майбутньому. Усі обчислення на графіку виконуються шляхом з'єднання тензорів;. [9]

- *NumPy* для високопродуктивних наукових обчислень і аналізу даних - це основний пакет наукових обчислень на Python. Ця бібліотека Python, яка надає об'єкт багатовимірного масиву, різні похідні об'єкти такі як замасковані масиви та матриці та набір підпрограм для швидких операцій над масивами, в т.ч математичні, логічні, маніпуляції фігурами, сортування,

вибір, введення/виведення, дискретні перетворення Фур'є, базова лінійна алгебра, базова статистика операції, випадкове моделювання та багато іншого. В основі пакета NumPy лежить об'єкт *ndarray*. Який інкапсулює *n* - вимірні масиви

однорідних типів даних, з багатьох операцій які виконуються в скомпільованому коді для забезпечення продуктивності. Є кілька важливих відмінностей між масивами NumPy і стандартними послідовностями Python:

Масиви NumPy мають фіксований розмір під час створення, на відміну від списків Python (які можуть динамічно зростати). Зміна розміру *ndarray* буде створити новий масив і видалити вихідний.

Усі елементи в масиві NumPy мають бути однаковими тип даних , отже, матиме той самий розмір у пам'яті.

Масиви NumPy полегшують розширені математичні та інші типи операції над великою кількістю даних. Як правило, такі операції виконуються більш ефективно та з меншою кількістю коду, ніж це можливо за допомогою вбудованою послідовністю Python.:[10]

- *Pandas* для аналізу даних загального призначення, бібліотека програмного забезпечення з відкритим вихідним кодом, створена на основі Python спеціально для обробки та аналізу даних, *Pandas* пропонує структуру даних і операції для потужного, гнучкого та легкого у використанні аналізу й обробки даних. *Pandas* покращує Python, надаючи цій мові програмування можливість працювати з даними, подібними до електронних таблиць, уможливлуючи швидке завантаження, вирівнювання, маніпулювання та об'єднання на додаток до інших ключових функцій.

Бібліотека з відкритим вихідним кодом *Pandas* включає *DataFrames*, які є двовимірними таблицями даних у вигляді масиву, у яких кожен стовпець містить значення однієї змінної, а кожен рядок містить один набір значень із кожного стовпця. Дані, що зберігаються в *DataFrame*, можуть бути числовим або символьним типом даних.[16]

- *Keras* бібліотека нейронної мережі з відкритим вихідним кодом, написана на Python, яка працює поверх TensorFlow. Вона розроблена модульною, швидкою і простою у використанні. Написана на Python і

використовується для полегшення впровадження нейронних мереж. Також підтримує численні серверні обчислення нейронної мережі.[11]

- Seaborn для візуалізації даних.

Scikit-learn містить різноманітні алгоритми класифікації, регресії та кластеризації, включаючи машини опорних векторів, випадкові ліси, посилення градієнта, k-середні і розроблений для роботи з числовими та науковими бібліотеками Python NumPy та SciPy. [17]

Середовище розробки Jupyter Notebook - додаток з відкритим вихідним кодом, в якому можна відразу побачити результат виконання коду.

Головна відмінність від традиційних інструментів розробки – можливість розбити код на частини та виконувати їх окремо. Наприклад, можна написати одну функцію і одразу перевірити, як вона працює, не запускаючи решту фрагментів коду. Також можна змінювати порядок виконання коду.

3.2. Аналіз даних фондових показників компанії

Пошуковий аналіз даних про акції компаній є важливим кроком у розумінні набору даних, і він може надати цінну інформацію про ефективність і тенденції акцій.

Date	Open	High	Low	Close	Adj Close	Volume
2018-05-15	79.389999	79.389999	78.261002	78.806000	78.806000	101550000
2018-05-16	78.875000	79.721497	78.833504	79.363998	79.363998	51412000
2018-05-17	79.028000	79.702003	78.650002	79.087997	79.087997	42952000
2018-05-18	79.066498	79.179497	78.605003	78.718498	78.718498	52852000
2018-05-21	79.250000	79.602501	78.750000	79.273003	79.273003	58504000

Рис 3.1. Фондові показники

З перших п'яти рядків видно, що дані для деяких дат відсутні, тому що у вихідні та святкові дні фондовий ринок залишається закритим, тому торгівля в ці дні не відбувається (рис 3.1).

За допомогою наступної функції (рис 3.2) переглянуто ненульові значення та типи даних в наборі, в результаті видно, що набір даних містить 1257 рядки та 7 стовпців. Стовпцями : є date, open, high, low, close, adj close і volume. Усі стовпці мають ненульові значення.

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1257 entries, 0 to 1256
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Date        1257 non-null   object
1   Open        1257 non-null   float64
2   High        1257 non-null   float64
3   Low         1257 non-null   float64
4   Close       1257 non-null   float64
5   Adj Close   1257 non-null   float64
6   Volume      1257 non-null   int64
dtypes: float64(5), int64(1), object(1)
memory usage: 68.9+ KB
```

Рис 3.2. Типи даних фондових показників

За допомогою функції *describe()* отримано базову статистику числових стовпців (рис 3.3).

	Open	High	Low	Close	Adj Close	Volume
count	1257.000000	1257.000000	1257.000000	1257.000000	1257.000000	1.257000e+03
mean	121.816898	123.312020	120.182598	121.759377	121.759377	8.306617e+07
std	33.763233	34.097431	33.356083	33.683698	33.683698	3.823255e+07
min	67.300003	69.801498	65.349998	67.197998	67.197998	1.762600e+07
25%	91.460500	92.327003	90.519997	91.461998	91.461998	5.743370e+07
50%	108.160004	111.180000	106.800003	108.789001	108.789001	7.217800e+07
75%	158.350006	159.937500	156.250000	158.100494	158.100494	9.787200e+07
max	187.199997	188.654007	184.839493	186.570496	186.570496	3.113460e+08

Рис 3.3. Статистика числових стовпців

В наборі даних відсутні пропущенні значення (рис 3.4):

```
df.isnull().sum()

Date      0
Open      0
High      0
Low       0
Close     0
Adj Close 0
Volume    0
dtype: int64
```

Рис 3.4 Перевірка відсутніх значень

3.3. Результати реалізації рекурентних нейронних мереж

RNN — мережева структура глибокого навчання. Перевага RNN полягає в тому, що вона враховує контекст даних під час навчання процесу.

Кожна нейронна мережа була розділена на тренувальний та тестовий набір де 80% даних використовуються для тренування, а 20% - для тестування моделі.

Для початку була побудована проста RNN яка складається з одного шару з 64 нейронами, після шару RNN додано шар *Dense* з одним нейроном. Модель компіюється з оптимізатором *adam* і функцією втрат *mean_squared_error*.

Кількість епох 100 (рис 3.5).

Показники оцінки моделі:

- Train MAE: 0.2886
- Train MSE: 0.1255
- Test MAE: 0.3016
- Test MSE: 0.1308

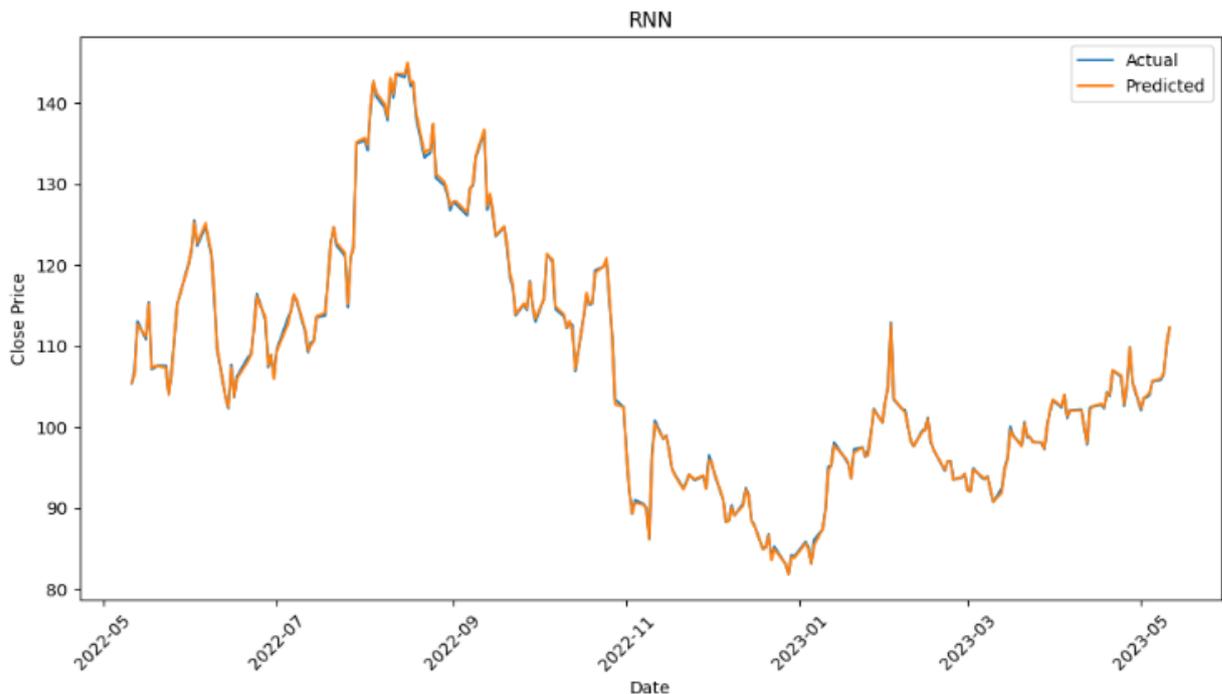


Рис 3.5 Рекурентна нейрона мережа

Наступна RNN побудована з двох шарів 64 нейронів кожна, компіляція та навчання моделі відповідна до попередньої.

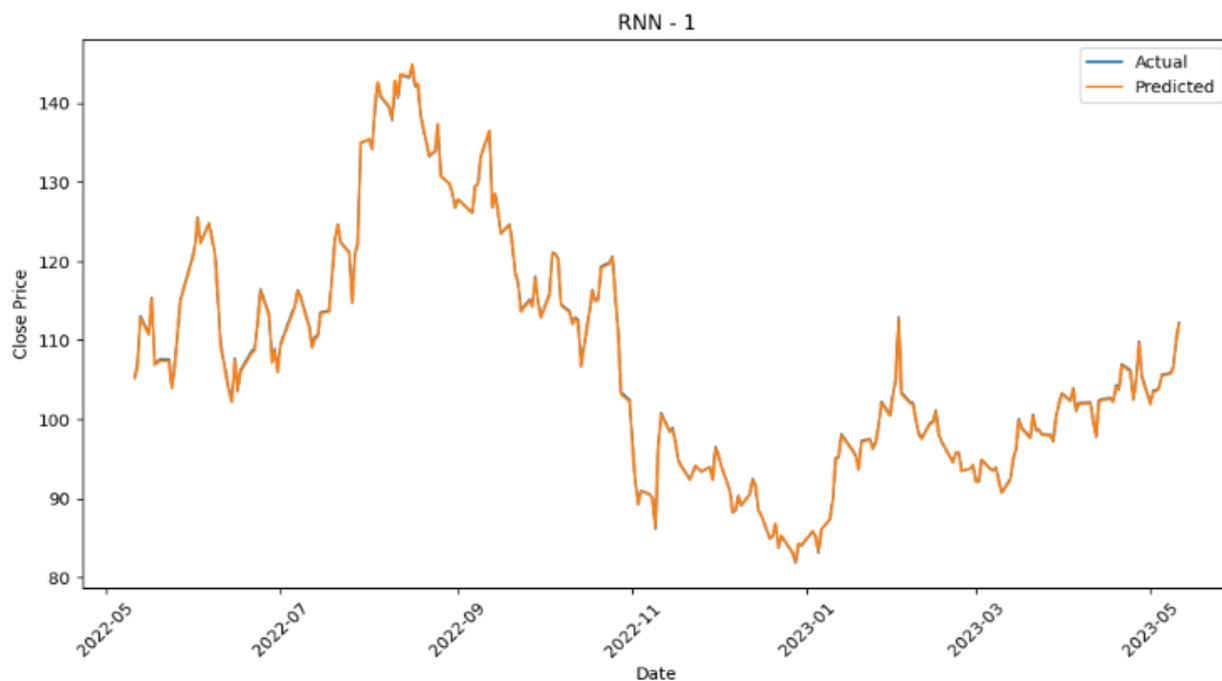


Рис 3.6 Графік прогнозів та фактичних значень

Результати навчання:

- Train MAE: 0.1081
- Train MSE: 0.0185
- Test MAE: 0.1242
- Test MSE: 0.0197

Наступна архітектура RNN складається з трьох шарів, перший з 128 нейронами, приймає послідовні входні дані та повертає послідовні вихідні дані, наступний шар з 64 нейронами який приймає входні дані з першого шару і повертає лише останній вихідний стан також додано шар Dense з 1 нейроном.

Компіляція та навчання моделі відповідна до попередніх (рис 3.7).

Показники оцінки моделі:

- Train MAE: 0.0429
- Train MSE: 0.0038
- Test MAE: 0.0718
- Test MSE: 0.0061

Зі збільшення кількості шарів та нейронів модель RNN показує точніші результати, остання модель показує найкращі результати з найменшими похибками MAE та MSE.

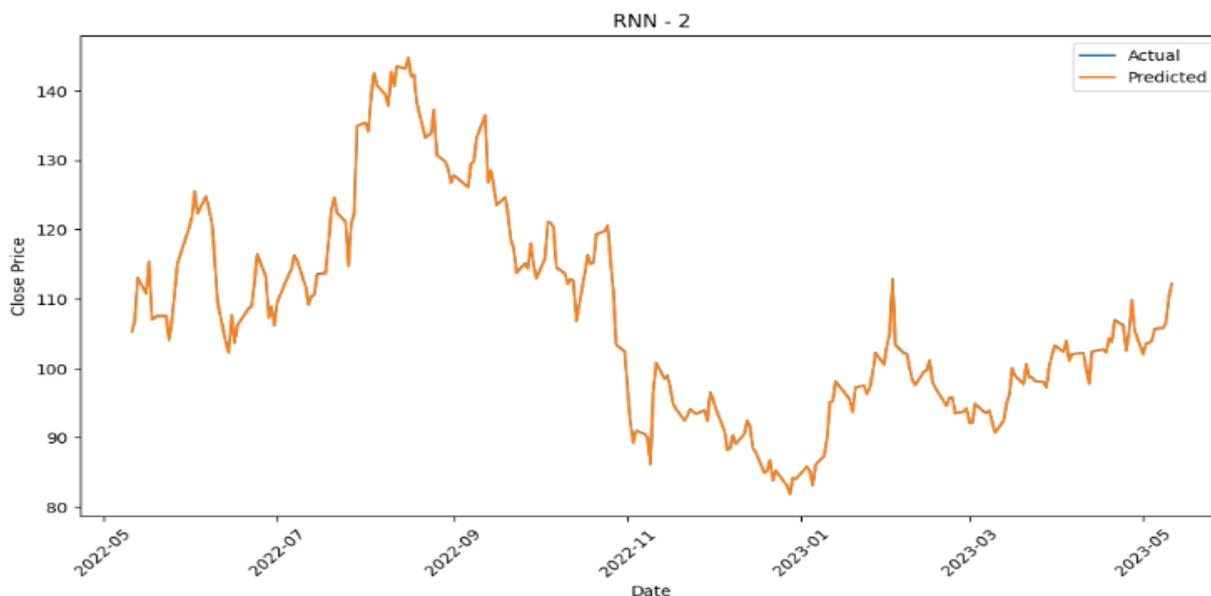


Рис 3.7 Графік прогнозів та фактичних значень

3.4. Результати реалізації довгої короткочасної пам'яті

LSTM, скорочення від Long Short-term Memory, є надзвичайно потужним алгоритмом для часових рядів. Він може фіксувати історичні тенденції та прогнозувати майбутні значення з високою точністю.

Кожна LSTM модель компілюється з оптимізатором *adam* і функцією втрат *mean_squared_error*. Кількість епох 200 (Табл 3.1).

Таблиця 3.1. Показники оцінки моделі LSTM

№	units	Train	Train	Test	Test
		MAE	MSE	MAE	MSE
1	64	0.7383	0.9647	1.0006	1.4501
2	128,64	0.0826	0.0137	0.0117	0.0220
3	64, 64	0.9887	2.3996	0.6743	0.7013
4	128	0.4454	0.3528	0.4491	0.3038

Найкращий результат LSTM з архітектурою: один шар з 128 нейронами цей шар повертає послідовні виходи для подальшого використання, другий шар з 64 нейронами який отримує послідовні виходи з попереднього та один шар *Dense* з одним виходом, для отримання остаточного прогнозу (рис 3.8).

Показники оцінки моделі:

- Train MAE: 0.0826
- Train MSE: 0.0137
- Test MAE: 0.0117
- Test MSE: 0.0220

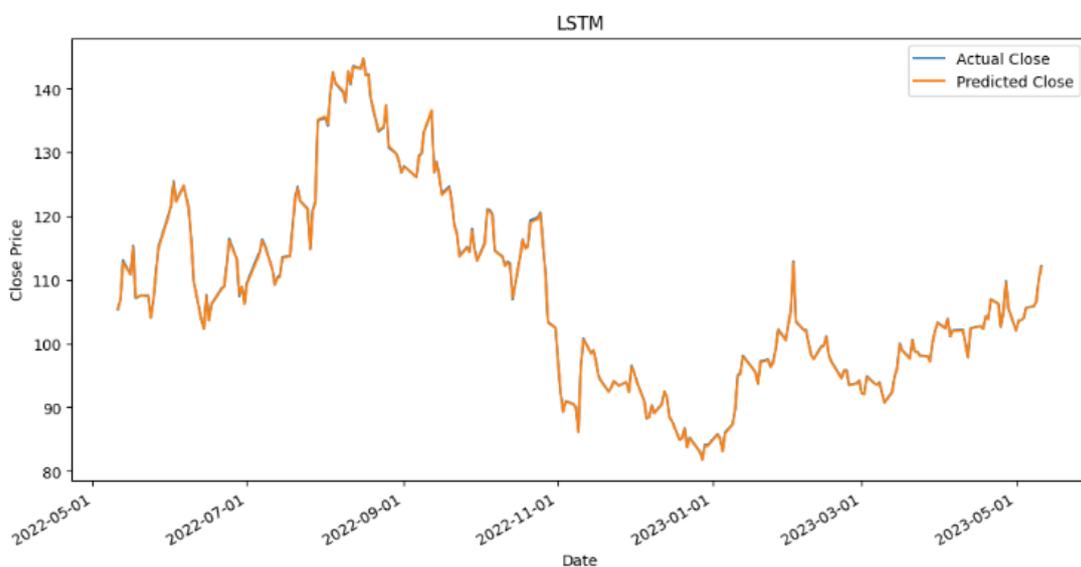


Рис 3.8 Графік прогнозів та фактичних значень LSTM

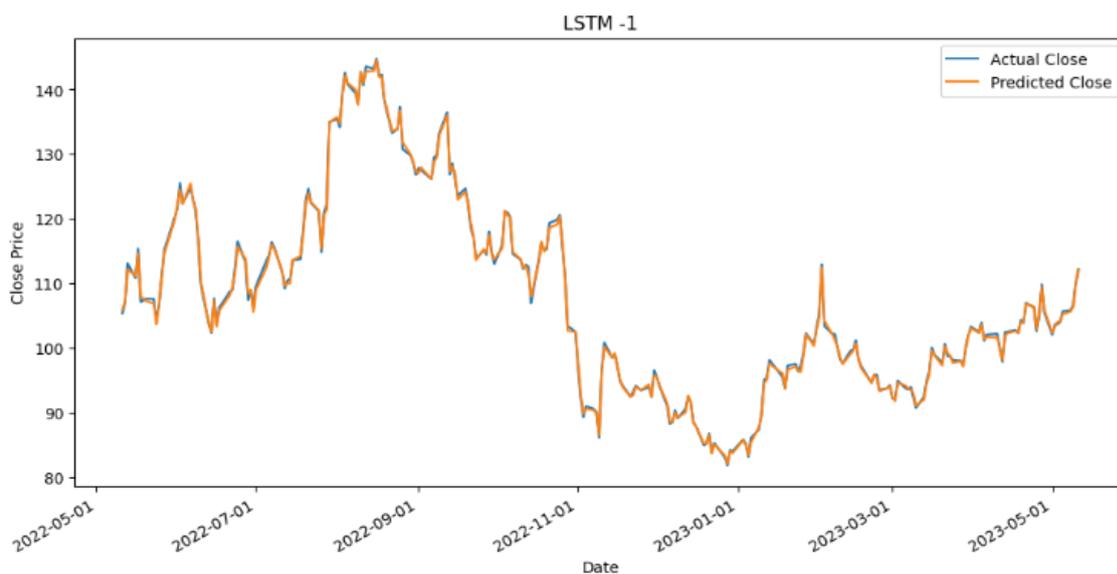


Рис 3.9 Прогноз та фактичні значення моделі LSTM

На Рисунку 3.9 LSTM яка складається з одного шару з 64 нейронами, після якого додано шар Dense з одним нейроном. Модель компілюється з оптимізатором *adam* і функцією втрат *mean_squared_error*.

Показники оцінки моделі:

- Train MAE: 0.7383
- Train MSE: 0.9647
- Test MAE: 1.0006
- Test MSE: 1.4501

Наступна LSTM з архітектурою: один шар з 64 нейронами який повертає послідовні виходи для подальшого використання, наступний з 64 нейронами який отримує послідовні виходи з попереднього та один шар Dense з одним виходом, для отримання остаточного прогнозу (рис 3.10).

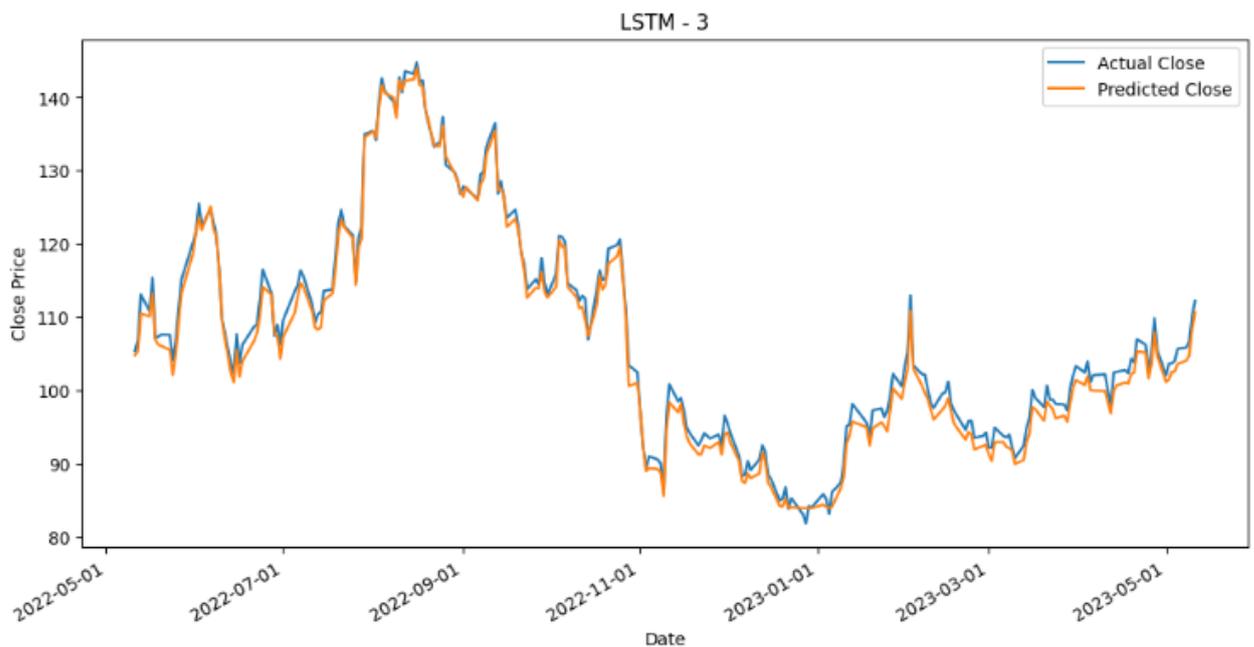


Рис 3.10 Графік прогнозів та фактичних значень LSTM

Результати навчання:

- Train MAE: 0.9887
- Train MSE: 2.3996
- Test MAE: 0.6743
- Test MSE: 0.3038

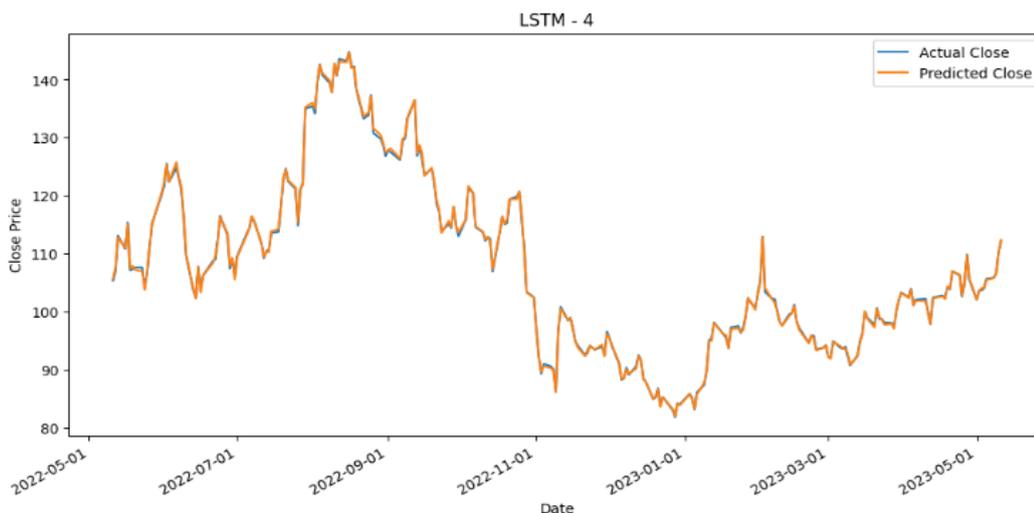


Рис 3.11 Графік прогнозів та фактичних значень LSTM

Показники оцінки моделі:

- Train MAE: 0.4454
- Train MSE: 0.3228
- Test MAE: 0.4491
- Test MSE: 0.3038

Остання LSTM (рис 3.11) яка складається з одного шару з 128 нейронами, після якого додано шар Dense з одним нейроном. Зі збільшенням кількості шарів нейронів LSTM показує точніші результати. При зменшенні кількості епох та нейронів метрика MSE та MAE збільшується.

3.5. Результати реалізації Багатошарового Персептрона

Багатошаровий персептрон (MLP) є однією з фундаментальних і ранніх нейронних мереж, також відомих як звичайні нейронні мережі. Її класифікують як мережу прямого зв'язку, що вказує на те, що дані передаються лише в одному напрямку, вперед, на відміну від інших нейронних мереж, які можуть обробляти послідовні дані.

Таблиця 2. Показники оцінки моделі MLP

№	units	epochs	Train	Train	Test	Test
			MAE	MSE	MAE	MSE

1	32	100	0.2294	0.0976	0.2711	0.1231
2	64	100	0.5496	0.5473	0.6909	0.7395
3	64	200	0.1971	0.0751	0.2331	0.0831
4	64, 64	100	0.0158	0.0004	0.0236	0.0011
5	128	100	0.2890	0.1590	0.3458	0.1827
6	128	200	0.0261	0.0015	0.0253	0.0009

MLP побудована з 6 різними архітектурами які мають різну кількість прихованих шарів. Модель компілюється з оптимізатором *adam* і функцією втрат *mean_squared_error*, функція активації *relu*.

Найкращий результат модель MLP має один прихований шар з 128 нейронами та використовує функцію активації ReLU. Останній шар має один нейрон без функції активації, що призначений для прогнозування ціни акції(рис 3.12).

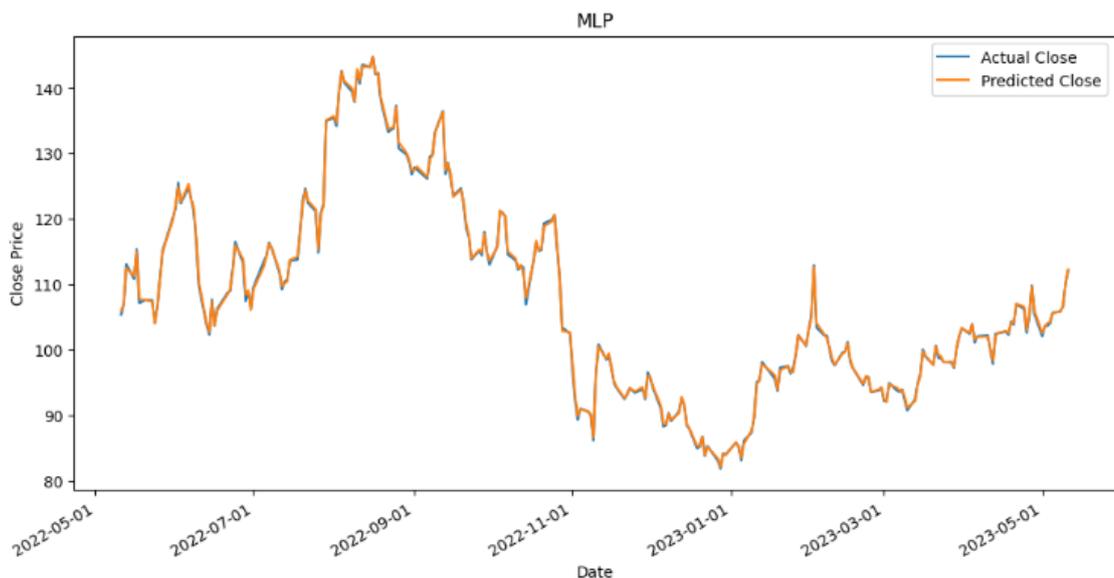


Рис 3.12 Графік прогнозів та фактичних значень MLP

Результати навчання MLP:

- Train MAE: 0.0261
- Train MSE: 0.0015
- Test MAE: 0.0253
- Test MSE: 0.0009

Наступна MLP модель має один прихований шар з 32 нейронами та використовує функцію активації ReLU. Кількість епох – 100 (рис 3.13).

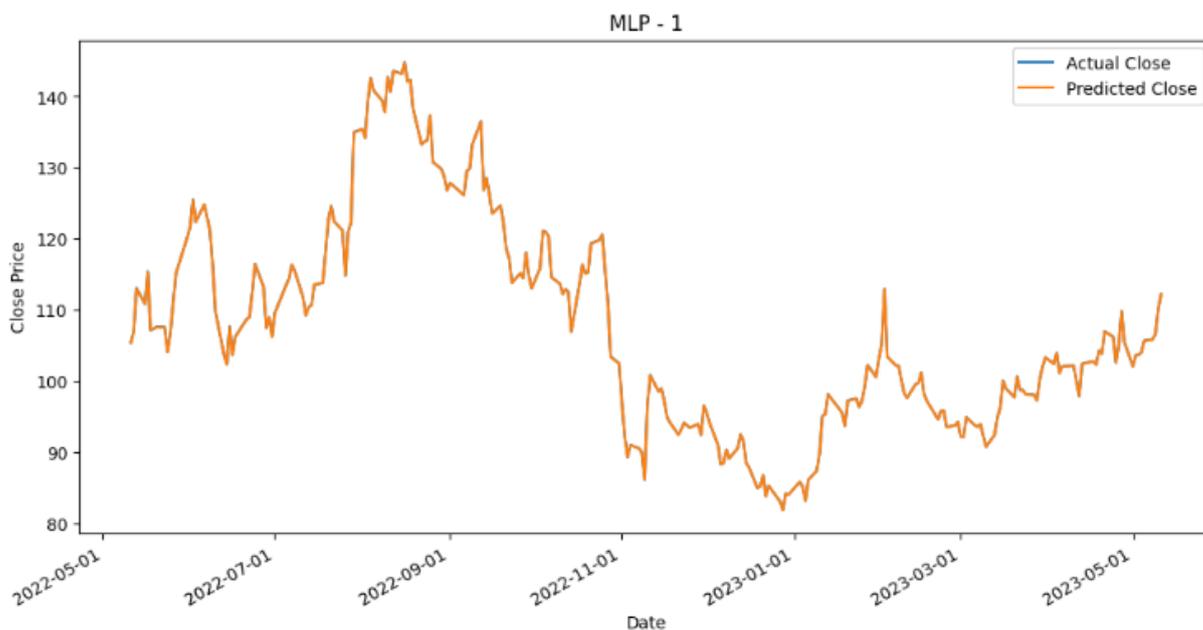


Рис 3.13 Графік прогнозів та фактичних значень MLP

Результати навчання MLP:

- Train MAE: 0.2294
- Train MSE: 0.0976
- Test MAE: 0.2711
- Test MSE: 0.1231

Друга та третя MLP модель має один прихований шар з 64 нейронами та використовує функцію активації ReLU. Кількість епох – 100 та 200 відповідно (рис 3.14), (рис 3.15).

Результати навчання MLP з одним прихованим шаром та кількістю епох 100:

- Train MAE: 0.5496
- Train MSE: 0.5473

- Test MAE: 0.6909

- Test MSE: 0.7395

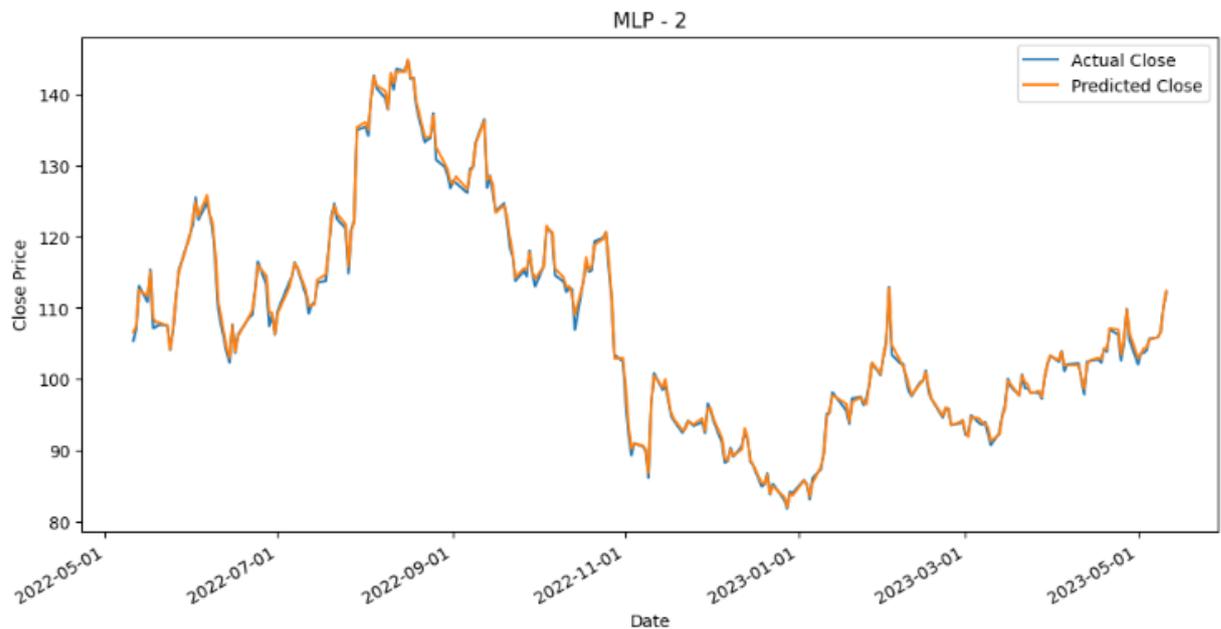


Рис 3.14 Прогнозовані та фактичні значення MLP

Результати навчання MLP з одним прихованим шаром та кількістю епох 200:

- Train MAE: 0.1971

- Train MSE: 0.0751

- Test MAE: 0.2331

- Test MSE: 0.0831

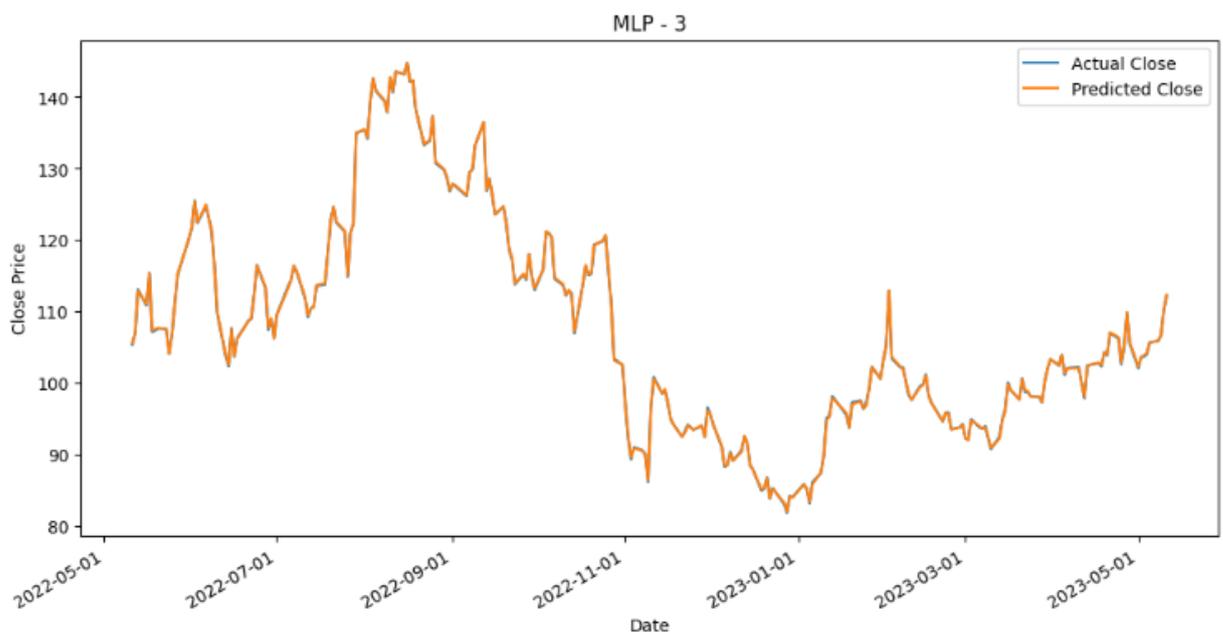


Рис 3.15 Прогнозовані та фактичні значення MLP

Наступна модель MLP має два шари з 64 нейронами кожна та використовує функцію активації ReLU. Останній шар має один нейрон без функції активації, що призначений для прогнозування ціни акції (рис 3.16).

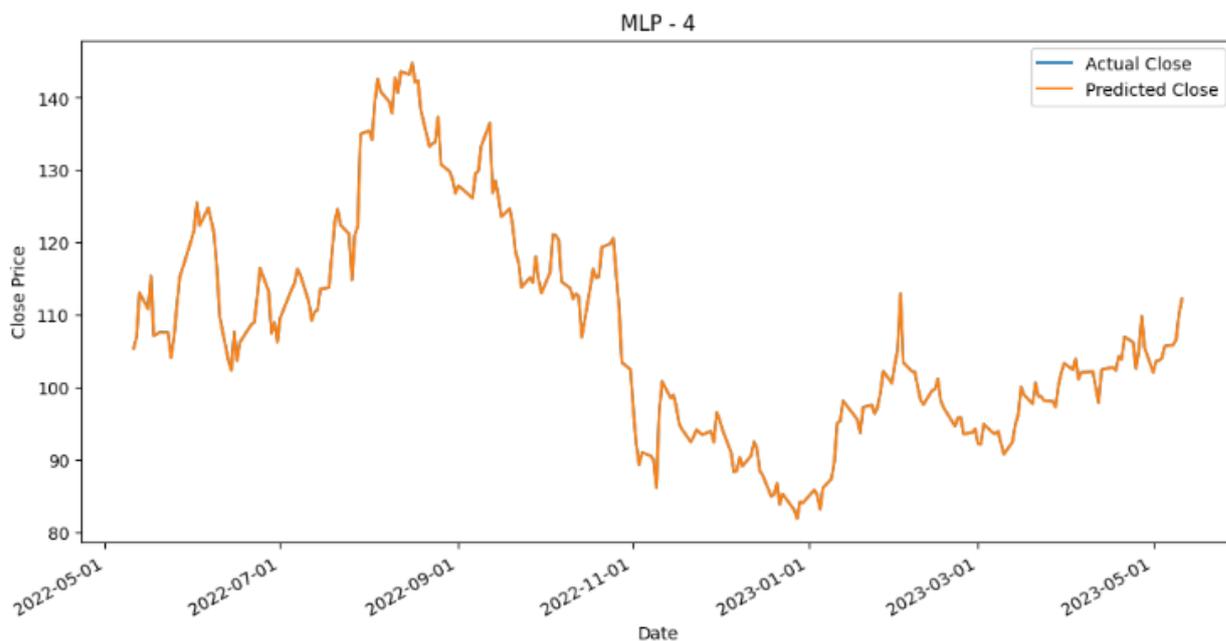


Рис 3.16 Прогнозовані та фактичні значення MLP

Результати навчання MLP:

- Train MAE: 0.0158
- Train MSE: 0.0004
- Test MAE: 0.0236
- Test MSE: 0.0011

Остання модель MLP має один прихований шар з 64 нейронами та використовує функцію активації ReLU. Кількість епох – 100 (рис 3.17).

Результати навчання MLP:

- Train MAE: 0.2890
- Train MSE: 0.1590
- Test MAE: 0.3458
- Test MSE: 0.1827

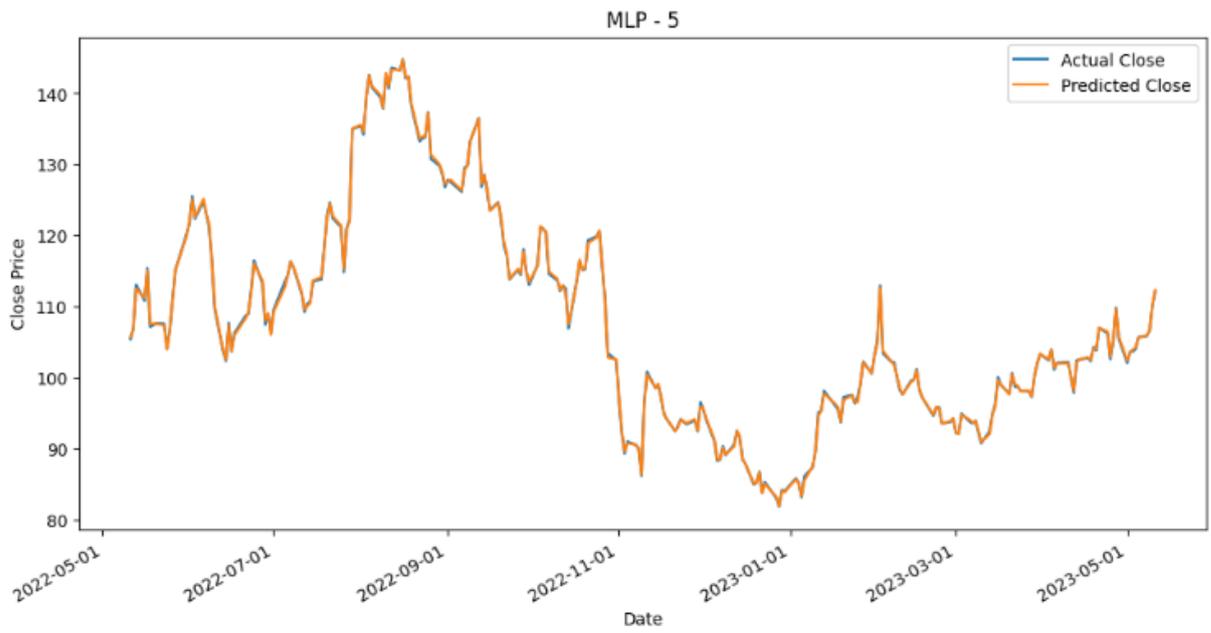


Рис 3.17 Прогнозовані та фактичні значення MLP

ВИСНОВКИ

В дипломній роботі досліджено методи прогнозування динаміки фондового ринку за допомогою штучних нейронних мереж. Прогнозування вартості цінних паперів за допомогою машинного навчання допомагає дізнатися ціну акції компанії та інших фінансових активів.

Створено програмну реалізацію моделей найбільш поширених нейронних мереж з різними архітектурами за допомогою мови програмування Python та в середовищі розробки Jupyter Notebook.

Для створення моделей були використанні: Рекурентні нейронні мережі, Модель довготривалої короткочасної пам'яті, Багатошаровий перцептрон.

Дані штучні нейронні мережі доцільно використовувати для прогнозування вартості цінних паперів, при достатніх наборах даних. Також моделі можна перенавчати під нові дані.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Gupta A, Dhingra B. Stock market prediction using hidden Markov models. In: 2012 Students Conference on Engineering and Systems. IEEE; 2012. pp. 1-4
2. Asadi S, Hadavandi E, Mehmanpazir F, Nakhostin MM. Hybridization of evolutionary Levenberg–Marquardt neural networks and data pre-processing for stock market prediction. Knowledge-Based Systems. 2012;35:245-258
3. Fundamental Analysis of Stocks: Explained. URL: <https://www.mstock.com/articles/fundamental-analysis> (дата звернення 21.06.2023)
4. Predicting Stock Prices Using Machine Learning. URL: <https://neptune.ai/blog/predicting-stock-prices-using-machine-learning> (дата звернення 15.06.2023)
5. Recent Advances in Stock Market Prediction Using Text Mining: A Survey Written By Faten Subhi Alzazah and Xiaochun Cheng URL: <https://www.intechopen.com/chapters/72381>(дата звернення 15.06.2023)
6. Applying machine learning in financial markets: A review of state-of-the-art methods. Nowadays, more and more trading firms are using machine learning technology to analyze the stock market..by Kerem Gülen January 11, 2023 URL:<https://dataconomy.com/2023/01/11/stock-prediction-machinelearning/> (дата звернення 16.06.2023)
7. Neurons in Neural Networks URL: <https://www.baeldung.com/cs/neural-networks-neurons> (дата звернення 16.06.2023)
8. What are recurrent neural networks? URL: <https://www.ibm.com/topics/recurrent-neural-networks>(дата звернення 20.06.2023)
9. What is TensorFlow? URL: <https://www.javatpoint.com/tensorflow-introduction> (дата звернення 20.06.2023)
10. What is NumPy? URL: <https://numpy.org/doc/stable/user/whatisnumpy.html>

(дата звернення 20.06.2023)

11. What is Keras and How it works? An Overview and Its Use Cases
Rajesh Kumar URL: <https://www.devopsschool.com/blog/what-is-keras-and-how-it-works-an-overview-and-its-use-cases/> (дата звернення 20.06.2023)
12. What Is a Time Series and How Is It Used to Analyze Data? URL: <https://www.investopedia.com/terms/t/timeseries.asp> (дата звернення 20.06.2023)
13. What Is Time-Series Forecasting? URL: <https://www.timescale.com/blog/what-is-time-series-forecasting/> (дата звернення 20.06.2023)
14. LSTM -RNN Model to Predict Future Stock Prices using an Efficient Optimizer Nikitha Saurabh
15. A Gentle Introduction to the Rectified Linear Unit (ReLU) By Jason Brownlee
URL: <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/> (дата звернення 20.06.2023)
16. Pandas URL: <https://www.nvidia.com/en-us/glossary/data-science/pandas-python/> (дата звернення 20.06.2023)
17. Why Use Python for AI and Machine Learning? URL: <https://steelkiwi.com/blog/python-for-ai-and-machine-learning/> (дата звернення 17.06.2023)
18. Long Short-Term Memory Networks (LSTM)- simply explained! URL: <https://databasecamp.de/en/ml/lstms/> (дата звернення 17.06.2023)
19. Alex Graves, Supervised Sequence Labelling with Recurrent Neural Networks

ДОДАТОК А

Реалізація моделі LSTM

```
import numpy as np
import pandas as pd
import tensorflow as tf

from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error
import matplotlib.pyplot as plt
from matplotlib.dates import DateFormatter

stock_data = pd.read_csv('./AMZN.csv')
sc = MinMaxScaler(feature_range=(0, 1))
sc_data = scaler.fit_transform(dataset)
sc_output = scaler.fit_transform(output)
X = sc_data
y = sc_output
X_train, X_test,
y_train, y_test = train_test_split(X, y,
test_size=0.2,
shuffle=False)
lstm = tf.keras.Sequential([
    tf.keras.layers.Lstm(128, activation='relu',
recurrent_dropout=0.2,
input_shape=(X_train.shape[1], X_train.shape[2])),
    tf.keras.layers.Dense(1)
])
```

```
lstm.compile(optimizer='adam',
loss='mean_squared_error')
history = lstm.fit(X_train, y_train,
epochs=200,
batch_size=32, validation_data=(X_test, y_test), verbose=1)
train_predictions = lstm.predict(X_train)
test_predictions = lstm.predict(X_test)
train_predictions = sc.inverse_transform
(train_predictions)
test_predictions = sc.inverse_transform
(test_predictions)
y_train = sc.inverse_transform(y_train)
y_test = sc.inverse_transform(y_test)
plt.figure(figsize=(12, 6))
plt.plot(dates_test, y_test, label='Actual Close')
plt.plot(dates_test, test_predictions, label='Predicted Close')
plt.xlabel('Date')
plt.ylabel('Close Price')
plt.title('LSTM Prediction')
plt.legend()
plt.show()
train_mae = mean_absolute_error
(y_train, train_predictions)
train_mse = mean_squared_error
(y_train, train_predictions)
mae = mean_absolute_error(y_test, test_predictions)
mse = mean_squared_error(y_test, test_predictions)
print(f"Train MAE: {train_mae:.4f}")
```

```
print(f"Train MSE: {train_mse:.4f}")  
print(f"Test MAE: {test_mae:.4f}")  
print(f"Test MSE: {test_mse:.4f}")
```

ДОДАТОК Б

Таблиця 3. Частина вхідних даних за 2018 рік

Date	Open	High	Low	Close	Adj Close
15.05.2018	79.389999	79.389999	78.261002	78.806000	78.806000
16.05.2018	78.875000	79.721497	78.833504	79.363998	79.363998
17.05.2018	79.028000	79.702003	78.650002	79.087997	79.087997
18.05.2018	79.066498	79.179497	78.605003	78.718498	78.718498
21.05.2018	79.250000	79.602501	78.750000	79.273003	79.273003
22.05.2018	79.494499	79.494499	78.762497	79.070000	79.070000
23.05.2018	78.552498	80.093002	78.317001	80.093002	80.093002
24.05.2018	79.901497	80.412003	79.418999	80.153503	80.153503
25.05.2018	80.150002	80.706001	80.022499	80.507500	80.507500
29.05.2018	80.035500	81.089500	80.007500	80.643501	80.643501
30.05.2018	80.904999	81.300003	80.646500	81.244499	81.244499
31.05.2018	81.150002	81.750000	81.067497	81.481003	81.481003
01.06.2018	81.851501	82.336502	81.754501	82.077003	82.077003
04.06.2018	82.445000	83.283997	82.274498	83.263496	83.263496
05.06.2018	83.649498	84.949997	83.502998	84.817497	84.817497
06.06.2018	85.225502	85.724998	84.323502	84.787498	84.787498
07.06.2018	84.928001	84.995003	83.805496	84.464996	84.464996
08.06.2018	84.056000	84.472000	83.650497	84.199501	84.199501
11.06.2018	84.075500	84.711998	84.029503	84.456001	84.456001
12.06.2018	84.650002	84.975502	84.575996	84.937500	84.937500
13.06.2018	85.140503	85.687500	85.005997	85.242996	85.242996
14.06.2018	85.674004	86.239998	85.443497	86.193001	86.193001
15.06.2018	85.699997	86.043503	85.426003	85.798500	85.798500
18.06.2018	85.313004	86.336998	85.127998	86.189499	86.189499
19.06.2018	85.452003	86.805496	85.019501	86.738998	86.738998
20.06.2018	87.125000	88.146500	87.068001	87.503998	87.503998
21.06.2018	88.000000	88.154999	85.877998	86.511002	86.511002
22.06.2018	87.130997	87.150002	85.595001	85.783501	85.783501
25.06.2018	85.125504	85.250000	82.315498	83.157501	83.157501
26.06.2018	83.618500	85.078499	83.167000	84.554497	84.554497
27.06.2018	85.405502	85.597504	83.000000	83.025497	83.025497
28.06.2018	83.626999	85.275002	83.058502	85.072502	85.072502
29.06.2018	85.849998	86.170502	84.716003	84.989998	84.989998
02.07.2018	84.135002	85.694504	83.903000	85.689003	85.689003
03.07.2018	86.197998	86.250000	84.624001	84.697998	84.697998
05.07.2018	85.268997	85.534500	84.107498	84.986504	84.986504
06.07.2018	84.800003	85.763496	84.583504	85.531502	85.531502
09.07.2018	86.202499	86.977997	85.811501	86.950996	86.950996
10.07.2018	86.926498	87.500000	86.550003	87.153503	87.153503
11.07.2018	86.899498	87.848000	86.699997	87.750000	87.750000
12.07.2018	88.225502	89.900002	88.109001	89.831001	89.831001
13.07.2018	90.196503	90.764999	89.761497	90.651497	90.651497

16.07.2018	91.097504	92.097504	90.722504	91.124496	91.124496
17.07.2018	90.578003	92.584503	89.869003	92.196503	92.196503
18.07.2018	92.400002	92.944000	91.563499	92.146004	92.146004
19.07.2018	91.473000	92.050003	90.563499	90.648499	90.648499
20.07.2018	91.250504	91.741997	90.502998	90.684998	90.684998
23.07.2018	90.610497	90.949997	88.499496	90.099998	90.099998
24.07.2018	91.450500	92.000000	90.469002	91.461998	91.461998
25.07.2018	91.464996	93.192001	91.132004	93.180496	93.180496
26.07.2018	91.949997	92.234001	90.224998	90.400002	90.400002
27.07.2018	93.802498	94.002502	90.326500	90.863503	90.863503
30.07.2018	91.366501	91.474998	88.301003	88.960999	88.960999
31.07.2018	89.324501	90.091499	86.966003	88.872002	88.872002
01.08.2018	89.199997	89.921997	88.801003	89.858498	89.858498
02.08.2018	89.438499	91.828003	89.300003	91.716499	91.716499
03.08.2018	91.887001	92.050003	91.074997	91.164497	91.164497
06.08.2018	91.290497	92.388496	90.945999	92.387497	92.387497
07.08.2018	92.726501	93.486000	92.313499	93.124001	93.124001
08.08.2018	93.050003	94.575500	92.724998	94.325996	94.325996
09.08.2018	94.099998	95.728500	93.874001	94.926003	94.926003
10.08.2018	94.425499	94.974998	93.910500	94.315002	94.315002
13.08.2018	94.925003	96.250000	94.683502	94.809998	94.809998
14.08.2018	95.969498	96.050499	95.000000	95.982498	95.982498
15.08.2018	95.477501	95.810501	93.489502	94.130997	94.130997
16.08.2018	95.196999	95.250000	94.177498	94.325996	94.325996
17.08.2018	94.290001	94.400002	92.777496	94.111000	94.111000
20.08.2018	94.528503	94.587502	93.303001	93.835503	93.835503
21.08.2018	94.000000	94.887497	93.720497	94.170998	94.170998
22.08.2018	93.832001	95.290001	93.832001	95.245003	95.245003
23.08.2018	95.358498	95.974998	95.038002	95.144997	95.144997
24.08.2018	95.525497	95.800499	95.126999	95.269501	95.269501
27.08.2018	95.750000	96.385002	95.463997	96.384003	96.384003
28.08.2018	96.886497	97.088997	96.441002	96.640999	96.640999
29.08.2018	97.672501	99.934502	97.446999	99.904999	99.904999
30.08.2018	99.871002	101.278503	99.345001	100.119003	100.119003
31.08.2018	100.349998	101.119003	100.237000	100.635498	100.635498
04.09.2018	101.324997	102.525002	100.650002	101.975502	101.975502
05.09.2018	101.905502	102.018997	99.494499	99.740997	99.740997
06.09.2018	100.325500	100.375000	96.760498	97.915497	97.915497
07.09.2018	96.935501	98.760002	96.867500	97.603500	97.603500
10.09.2018	98.550003	98.652000	96.575996	96.950500	96.950500
11.09.2018	96.413498	99.444000	95.849998	99.357498	99.357498
12.09.2018	99.699997	100.000000	98.122002	99.500000	99.500000
13.09.2018	100.000000	100.438004	99.101501	99.493500	99.493500
14.09.2018	99.646500	99.682503	97.960999	98.509499	98.509499
17.09.2018	97.736504	97.841003	94.370499	95.401497	95.401497
18.09.2018	95.932503	97.910004	95.772003	97.052498	97.052498
19.09.2018	97.025002	97.041496	95.245003	96.320999	96.320999

20.09.2018	96.929001	97.750000	96.612503	97.214996	97.214996
21.09.2018	97.710999	97.865501	95.525002	95.750504	95.750504
24.09.2018	95.189499	96.844002	93.250000	96.718002	96.718002
25.09.2018	97.144997	98.795502	96.942497	98.727501	98.727501
26.09.2018	98.425003	99.762497	98.075996	98.742500	98.742500
27.09.2018	99.662003	100.807999	99.429001	100.649002	100.649002
28.09.2018	100.220497	101.325996	99.822998	100.150002	100.150002
01.10.2018	101.099503	101.659500	100.180000	100.218002	100.218002
02.10.2018	99.999496	100.669502	98.288498	98.565498	98.565498
03.10.2018	99.084999	99.485001	97.490501	97.638000	97.638000
04.10.2018	97.449997	97.800003	94.828499	95.471001	95.471001
05.10.2018	95.899498	96.454002	93.141502	94.482498	94.482498
08.10.2018	93.699997	95.099998	91.532997	93.221001	93.221001
09.10.2018	92.999496	94.834000	92.615997	93.515999	93.515999
10.10.2018	92.894501	92.928001	87.720497	87.762497	87.762497
11.10.2018	86.199997	87.769997	84.254997	85.968002	85.968002
12.10.2018	90.400002	90.447502	87.126503	89.430496	89.430496
15.10.2018	89.750000	89.752502	86.711502	88.047501	88.047501
16.10.2018	89.175003	91.194000	88.077499	90.998001	90.998001
17.10.2018	92.139503	92.250000	90.349998	91.586502	91.586502
18.10.2018	91.074501	91.507500	88.393501	88.536003	88.536003
19.10.2018	89.258003	90.455002	87.650002	88.201500	88.201500
22.10.2018	89.199997	90.474998	87.800003	89.464996	89.464996
23.10.2018	87.112000	88.817001	85.699997	88.434998	88.434998
24.10.2018	88.684998	88.885498	82.828003	83.209999	83.209999
25.10.2018	85.167000	89.740501	84.600502	89.108498	89.108498
26.10.2018	82.479500	84.922997	80.150002	82.140503	82.140503
29.10.2018	83.000000	83.287003	74.750000	76.944000	76.944000
30.10.2018	74.307999	77.049500	73.818001	76.521004	76.521004
31.10.2018	78.499496	81.195503	78.254501	79.900497	79.900497
01.11.2018	81.176498	83.522499	79.921997	83.276497	83.276497
02.11.2018	83.929497	84.872002	82.591499	83.276497	83.276497
05.11.2018	82.878502	82.904503	79.818001	81.389999	81.389999
06.11.2018	80.917503	83.250000	80.727501	82.140503	82.140503
07.11.2018	83.650002	87.961502	83.204002	87.774498	87.774498
08.11.2018	87.750000	89.199997	86.255501	87.745499	87.745499
09.11.2018	86.625000	87.195999	85.093498	85.621498	85.621498
12.11.2018	84.912003	85.427498	81.500504	81.842499	81.842499
13.11.2018	82.464500	83.852997	80.687500	81.558502	81.558502
14.11.2018	82.816002	83.650002	79.853500	79.950500	79.950500
15.11.2018	79.050499	81.240997	77.325500	80.972000	80.972000
16.11.2018	79.375000	80.723999	78.655998	79.670502	79.670502
19.11.2018	78.850502	79.059502	75.167999	75.614502	75.614502
20.11.2018	71.875000	76.737503	71.000000	74.773003	74.773003
21.11.2018	77.149498	77.500000	75.750000	75.836502	75.836502
23.11.2018	75.849998	76.809998	75.090500	75.102997	75.102997
26.11.2018	76.949997	79.240501	76.210999	79.066498	79.066498

27.11.2018	78.799500	79.882500	77.900497	79.070999	79.070999
28.11.2018	80.695999	84.072502	80.060997	83.887497	83.887497
29.11.2018	83.749496	84.499496	82.616501	83.678497	83.678497
30.11.2018	83.974998	84.800003	83.324997	84.508499	84.508499
03.12.2018	88.473000	88.917000	86.500000	88.617996	88.617996
04.12.2018	87.800003	88.516998	83.250000	83.419998	83.419998
06.12.2018	80.743500	85.052498	80.492500	84.959503	84.959503
07.12.2018	85.253502	85.946503	81.273003	81.456497	81.456497
10.12.2018	81.192001	82.899498	79.543503	82.051498	82.051498
11.12.2018	83.900002	83.973503	80.980003	82.162003	82.162003
12.12.2018	83.449997	85.249496	83.013496	83.177002	83.177002
13.12.2018	84.000000	84.606003	82.074997	82.918999	82.918999
14.12.2018	81.900002	82.128502	79.250000	79.595497	79.595497
17.12.2018	78.300003	78.806503	75.250504	76.045502	76.045502
18.12.2018	77.000000	78.377502	76.150497	77.573997	77.573997
19.12.2018	77.152496	79.226501	74.158997	74.753998	74.753998
20.12.2018	74.199997	75.474998	71.634499	73.041496	73.041496
21.12.2018	73.249496	74.000000	68.197998	68.872498	68.872498
24.12.2018	67.300003	69.801498	65.349998	67.197998	67.197998
26.12.2018	68.444504	73.657997	68.150497	73.544998	73.544998
27.12.2018	72.709999	73.449997	69.515503	73.082001	73.082001
28.12.2018	73.667503	75.673500	72.449997	73.901001	73.901001
31.12.2018	75.540001	76.038002	74.349998	75.098503	75.098503