

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра системного аналізу та теорії прийняття рішень

**Кваліфікаційна робота
на здобуття ступеня бакалавра**

за освітньо-професійною програмою «Системний аналіз»

за спеціальністю 124 Системний аналіз

на тему:

**Аспекти розробки інтелектуальної інформаційної системи для оцінки
вартості акцій фондового ринку на прикладі ринку цінних паперів США**

Виконав студент 4-го курсу
Костянтин ПЛАВОВ



Науковий керівник:
асистент, кандидат фіз.-мат. наук
Шевчук Юлія Михайлівна



Засвідчую, що в цій роботі немає запозичень з праць
інших авторів без відповідних посилань.

Студент



Роботу розглянуто й допущено до захисту на засіданні
кафедри системного аналізу та теорії прийняття
рішень

« 07 » _____ 06 _____ 2022 р., протокол № 10

Завідувач кафедри
Олександр НАКОНЕЧНИЙ



Київ – 2022

Зміст

Зміст	2
Вступ	4
1. Інтелектуальні інформаційні системи	7
1.1 Типова структура та склад інформаційних систем	7
1.2 Стадії та етапи розробки автоматизованої ІС	10
1.3 Забезпечення функціонування та розвитку АІС на підприємстві	16
2. Огляд інформаційних систем для фондових ринків	20
2.1 Використання штучного інтелекту на фондових ринках	21
2.2 Як використання штучного інтелекту на фондових ринках може сказатись на сфері фінансів і не тільки	23
2.2.1 Штучний інтелект та рекордна прибутковість інвестицій	23
2.2.2 Чому штучний інтелект скоро витіснить людей із біржі	25
2.2.3 Як це працює на практиці: 8 ІІ-компаній у сфері інвестицій	26
3. Нейронні мережі та алгоритми прогнозування ціни	29
3.1 Класифікація штучних нейронних мереж	30
3.2 Одношарові штучні нейронні мережі	31
3.3 Навчання по дельта-правилу	32
3.4 Багатошарові нейронні мережі	33
3.5 Навчання методом зворотного розповсюдження помилки	34
3.6 Завдання, які вирішуються нейронними мережами	36

	3
3.7 LSTM Модель	38
3.7.1 Опис гіперпараметрів LSTM Моделі.....	39
3.8 Модель Facebook Prophet.....	40
3.8.1 Гіперпараметри у Facebook Prophet's.....	41
4. Розробка інформаційної системи для прогнозування ціни акцій.....	43
4.1. Опис структури вхідних даних	43
4.2 Основні принципи обробки даних для інформаційної системи	44
4.3 Генерація прогнозу та різниця між запропонованими моделями	44
4.4 Основні принципи проведення тестів	45
4.5 Інтерфейс користувача.....	48
Висновки.....	51
Список використаних джерел	51
Додаток.....	55

Вступ

Оскільки сучасний світ не стоїть на місці, з кожним днем з'являються нові технології, методи та інновації, які оптимізують системи і процеси та роблять життя людей простішим .

Сьогодні запровадження новітніх технологій у сфері діяльності людини неможливо уявити без використання комп'ютерної та іншої телекомунікаційної техніки. Контролювати роботу підприємства, замовити товар, купити акції компанії, викликати таксі, оформити кредит в банку – це все можна зробити користуючись спеціальними додатками та інтернет-сервісами. У наш час стрімко набирають популярності автоматизовані інформаційні системи, які мають великі перспективи в різних сферах нашого життя. Що ж представляє собою ця система та які є загальні алгоритми при конструюванні таких систем?

Багато людей мріє побудувати кар'єру на фондових ринках але навіть не уявляють з якими конкурентно спроможними інформаційними системами вони зіштовхнуться на своєму шляху, моделями, які прораховують сотні варіантів менше ніж за хвилину.

Величезна кількість експертів впевнена, що майбутнє фондових ринків саме за штучним інтелектом. Світ вже переживав подібну ситуацію у сфері шахів. Першого гросмейстера штучний інтелект переміг ще у 1997 році, з тих пір розвиток штучного інтелекту пішов дуже стрімко, для прикладу зараз мобільний додаток по шахам переможе будь-кого, якщо грати на максимальному рівні [1]. Але це не призвело до втрати популярності шахів серед людей. З фондовими ринками трохи інша ситуація , в тому плані, що штучний інтелект буде працювати на одному і тому самому ринку, де працюють люди, від новачків до зірок з Wall-Street.

За для практичної реалізації проекту було взято дані деяких компаній з ринку цінних паперів США та була проведена спроба розробити свій штучний інтелект, який зможе прогнозувати ціну акцій і було проведено декілька тестів на даних з минулого, які показали, що розроблений штучний інтелект може бути дуже

прибутковим у деякі періоди часу.

Для тестування і валідації роботи було обрано список компаній і індексів які мають низьку кореляцію між собою. Зроблено це було для отримання різних даних для подальшого аналізу і виявлення деяких закономірностей в залежності від прибутковості тестів на різних цінних паперах.

Мета роботи:

- Розробка ефективної інформаційної системи для застосування на фондових ринках.
- Реалізація веб-порталу для роботи користувача з інформаційною системою.

Завдання:

- Дослідження існуючих підходів для реалізації інформаційних систем.
- Аналіз методів прогнозування ціни акцій.
- Обґрунтування ефективності запропонованих методів реалізації інформаційної системи.
- Проведення тестів прибутковості запропонованих методів
- Інтеграція запропонованої інформаційної системи у веб-портал.

Об'єкт дослідження: веб-портал для прогнозування ціни акцій двома методами як сервіс.

Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновку, списку використаної літератури. Список використаної літератури включає в себе 21 джерело. Робота виконана на 64 сторінках друкованого тексту.

У першому розділі розглянуто поняття інтелектуальних інформаційних систем, проаналізовано основні їх функції. Проаналізовано різні підходи розробки таких систем, які можуть застосовуватися в залежності від наявності даних, предметної області. Досліджено основні принципи побудови інформаційних систем.

У другому розділі проаналізовано основні методи використання

інформаційних систем на фондовому ринку, наведено плюси і мінуси використання штучного інтелекту як частини інформаційної системи для фондового ринку.

У третьому розділі досліджено методи побудови і навчання нейронних мереж, описані два методи які використовуються у практичній реалізації проекту.

У четвертому розділі запропоновано архітектуру інтелектуальної інформаційної системи. Описано основні корки програмної реалізації системи, а також її інтеграцію у інтерактивний веб-портал.

1. Інтелектуальні інформаційні системи

1.1 Типова структура та склад інформаційних систем

Структуру інформаційної системи складає сукупність окремих її частин, які називаються підсистемами. Підсистема - це частина системи, які виділена за будь-якою ознакою.

Практично всі різновиди інформаційних систем незалежно від сфери їх застосування включають один і той самий набір компонентів (Рис. 1.1) :

- функціональні;
- система опрацювання даних;
- організаційні компоненти.



Рисунок 1.1 - Декомпозиція інформаційної системи.

Під функціональними компонентами мають на увазі систему функцій управління — повний набір взаємопов'язаних у часі й просторі робіт з управління, необхідних для досягнення поставлених перед підприємством цілей.

Залежно від складності конкретного підприємства кількість функціональних підсистем коливається від 5 до 50 найменувань.

Вибір складу функціональних задач функціональних підсистем управління здійснюється звичайно з урахуванням основних фаз управління: планування; обліку, контролю і аналізу; регулювання (виконання).

Планування — це управлінська функція, що забезпечує формування планів, відповідно до яких буде організоване функціонування об'єкта управління. Традиційно виділяють перспективне (5-10 років), річне (1 рік) і оперативне (доба, тиждень, декада, місяць) планування.

Облік, контроль і аналіз — функції, що забезпечують одержання даних про стан керованої системи за певний проміжок часу, визначення факту та причини відхилення фактичного стану об'єкта управління від його планованого стану, а також виявлення розмірів цього відхилення. Облік ведеться за показниками плану в обраному діапазоні планування (оперативний, середньостроковий і інші).

Регулювання (виконання) — це функція, що забезпечує порівняння планованих та фактичних показників функціонування об'єкта управління і реалізацію необхідних керуючих впливів за наявності відхилень від запланованих у заданому діапазоні (відрізку). Відповідно до виділених функціональних підсистем та з урахуванням вимог управління і визначається склад задач функціональних підсистем. Наприклад, інформаційна система управління персоналом підприємства може містити такі функціональні підсистеми [2]:

- планування чисельності персоналу підприємства;
- розрахунок фонду заробітної плати персоналу;
- планування та організація навчання персоналу;
- управління кадровими переміщеннями;
- статистичний облік і звітність;
- довідки за запитом.

Система опрацювання даних (СОД) призначена для інформаційного обслуговування фахівців різних органів управління підприємства, що приймають управлінські рішення.

Практично всі системи опрацювання даних інформаційних систем незалежно від їх сфери застосування включають один і той самий набір складових компонентів, що називаються видами забезпечення. Прийнято виділяти інформаційне, програмне, технічне, правове, лінгвістичне забезпечення.

Інформаційне забезпечення — це сукупність методів і засобів розміщення й організації інформації, що включають у себе системи класифікації і кодування, уніфіковані системи документації, раціоналізації документообігу та форми документів, методів створення внутрішньо машинної інформаційної бази інформаційної системи. Від якості розробленого інформаційного забезпечення значною мірою залежить достовірність і якість прийнятих управлінських рішень [3].

Програмне забезпечення — сукупність програмних засобів для створення та експлуатації СОД засобами обчислювальної техніки. До складу програмного забезпечення входять базові (загальносистемні) та прикладні (спеціальні) програмні продукти.

Базові програмні засоби служать для автоматизації взаємодії людини і комп'ютера, організації типових процедур опрацювання даних, контролю і діагностики функціонування технічних засобів СОД.

Прикладне програмне забезпечення - це сукупність програмних продуктів, призначених для автоматизації вирішення функціональних задач інформаційної системи. Вони можуть бути розроблені як універсальні засоби (текстові редактори, електронні таблиці, системи управління базами даних) і як спеціалізовані, тобто такі, що реалізують функціональні підсистеми (бізнес-процеси) об'єктів різної природи (економічні, інженерні, технічні тощо).

Технічне забезпечення - це комплекс технічних засобів, що застосовуються для функціонування системи опрацювання даних, і містить у собі пристрої, за

допомогою яких виконуються типові операції опрацювання даних як поза ПК (периферійні технічні засоби збору, реєстрації, первинного опрацювання інформації, оргтехніка різного призначення, засоби телекомунікації і зв'язку), так і на ПК різних класів.

Правове забезпечення — це сукупність правових норм, що регламентують створення і функціонування інформаційної системи. Правове забезпечення розробки інформаційної системи включає нормативні акти договірних взаємовідносин між замовником і розроблювачем ІС, правове регулювання відхилень. Правове забезпечення функціонування СОД включає: умови надання юридичної чинності документам, отриманим із застосуванням обчислювальної техніки; права, обов'язки і відповідальність персоналу, в тому числі за своєчасність і точність опрацювання інформації; правила користування інформацією і порядок вирішення суперечок щодо її достовірності.

Лінгвістичне забезпечення — це сукупність мовних засобів, що використовуються на різних стадіях створення та експлуатації СОД для підвищення ефективності розробки й забезпечення спілкування людини і ПК.

Під організаційними компонентами ІС - це сукупність методів і засобів, які регламентують взаємодію робітників з технічними засобами та між собою в процесі розробки та експлуатації інформаційної системи [2].

1.2 Стадії та етапи розробки автоматизованої ІС

Стадії та етапи розробки інформаційних систем визначає відповідний державний стандарт, де наводиться повний перелік стадій та етапів створення інформаційних систем, причому в конкретних умовах ці стадії та етапи можуть поєднуватись один з одним або не виконуватись. Це залежить від особливостей інформаційних систем, які створюються, та від домовленості між розробником системи та її замовником.

У державному стандарті виділено вісім стадій створення інформаційних систем [2]:

- формування вимог до інформаційної (автоматизованої) системи;
- розробка концепції ІС;
- технічне завдання;
- ескізний проект;
- технічний проект;
- робоча документація;
- введення в експлуатацію;
- супроводження ІС.

На першому етапі проводиться обстеження об'єкта та обґрунтовується необхідність створення ІС, формулюються вимоги користувача до ІС, оформляється звіт про виконану роботу.

Під час обстеження об'єкта з'ясовується документообіг (у тому числі кількість документів за певний період часу), форми початкових та вихідних документів, методики розрахунку окремих показників.

Обстеження має виявити проблеми, розв'язання яких можливе засобами обчислювальної техніки, та надати оцінку доцільності створення ІС.

Обстеження проводиться шляхом бесід та консультацій із працівниками установи, для якої буде створюватись інформаційна система.

В окремих випадках може провадитись самохронометраж роботи.

На першому етапі разом із замовником погоджуються вимоги до ІС. Серед вимог можуть бути суми максимальних витрат на розробку, термін виконання розробки, умови функціонування системи, перелік функцій, які система має забезпечити, та ін [6].

Звіт про обстеження складається в довільній формі. На його підставі надалі розроблятиметься технічний проект, тому бажано в додатках до звіту навести форми використовуваних документів. У ньому ж необхідно викласти погоджені із замовником методики розрахунку економічних показників.

Вимоги до системи можуть бути оформлені як окремий документ. Для такого документа немає стандартної назви, але здебільшого він називається заявкою на розробку або тактико-технічне завдання.

Під час розробки концепції ІС (другий етап) проводяться науково-дослідні роботи для пошуку шляхів та оцінки можливостей реалізації вимог користувача. На цьому етапі можна визначити методи, які будуть покладені в основу розрахунків, або принципові підходи до розв'язування конкретних задач. Наприклад, для інформаційної системи, яка пов'язана з оптимальним плануванням виробництва, на цьому етапі можуть визначатися математичні моделі та методи (лінійне програмування, імітаційне моделювання тощо) для використання в розрахунках та стандартні пакети програмних засобів, які можна буде використати.

Цей етап закінчується складанням та затвердженням звіту про науково-дослідну роботу. Він може містити оцінку необхідних для реалізації ресурсів розробки та самої ІС, давати порівняльну характеристику тих чи інших варіантів розробки інформаційної системи, визначати порядок оцінки якості системи.

На третьому етапі формується технічне завдання (ТЗ) на створення АІС. ТЗ є основним документом, що визначає вимоги та порядок створення (розвитку або модернізації) автоматизованої системи. На підставі технічного завдання провадиться розробка інформаційної системи, її прийом в експлуатацію. ТЗ розробляють на систему в цілому. Додатково можуть бути розроблені ТЗ на окремі частини ІС.

Четвертий етап. На етапі розробки ескізного проекту опрацьовують попередні проектні рішення щодо всієї системи або її частин. Може бути визначений перелік задач, які будуть розв'язуватися в системі, концепція інформаційної бази, яка створюється, функції та параметри основних програмних засобів. Для кожної задачі в ескізному проекті можуть бути наведені погоджені із замовником форми первинних та вихідних документів, структури інформаційних масивів або їх перелік, основні алгоритми обробки інформації.

П'ятий етап. Етап розробки технічного проекту передбачає розробку проектних рішень щодо системи та її частин, розробку документації на АІС, розробку документації на постачання виробів для комплектації АІС або технічних вимог для їх розробки, розробку завдань на проектування суміжних частин проекту.

Проектні рішення щодо системи та її частин визначають її організаційну структуру, функції персоналу в АІС, структуру технічних засобів, мови програмування або СУБД, які використовуватимуться, наводять загальні характеристики програмного забезпечення, систем класифікації та кодування(визначають загальнодержавні або галузеві класифікатори, що їх необхідно використовувати), визначають варіанти ведення інформаційної бази.

Шостий етап. На етапі розробки документації на АІС визначаються державними стандартами. Обов'язково розробляється постановка задачі, алгоритм її розв'язання, описується інформаційне забезпечення (організація інформаційної бази, системи класифікації та кодування, інформаційні масиви), організаційне, технічне та програмне забезпечення. Усі ці проектні документи можуть оформлятися як окремі документи, а можуть входити у технічний проект як окремі розділи.

Документація на постачання виробів для комплектації ІС складається тоді, коли в установі не використовувалися засоби обчислювальної техніки або існуючих засобів недостатньо для обробки інформації. У такій документації, яка складається в довільній формі, обґрунтовується закупівля тих чи інших засобів та наводяться їх можливі закупівельні ціни. Наприклад, вибираються комплектуючі частини для ПК: обсяг оперативної пам'яті, ємність магнітного диска, характеристики принтера тощо.

Технічне завдання на розробку технічних засобів необхідне лише тоді, коли для обробки інформації потрібне нестандартне обладнання, яке не випускається промисловістю. Наприклад, для створення автоматизованої системи з метою обліку роботи депутатів Верховної Ради були замовлені спеціальні пристрої для реєстрації

депутатів та голосування, а також спеціальні табло, де відображуються результати голосування та інша інформація .

Розробка завдань на проектування в суміжних частинах проекту виконується тоді, коли для впровадження інформаційної системи необхідно виконати ряд підготовчих робіт, пов'язаних із будівельними, електротехнічними та іншими роботами. Ці завдання можуть бути довільної форми або подаватися згідно з вимогами до розробки документації в тій чи іншій галузі діяльності.

Під час створення робочого проекту формуються документи, які визначає стандарт для цього етапу проектування, та розробляються або адаптуються програми обробки інформації. Серед документів робочого проекту можуть бути: загальний опис системи, опис технологічного процесу обробки інформації, інструкції з виконання окремих операцій технологічного процесу, керівництво користувача, опис програм тощо.

Найважливішим під час створення робочого проекту є розробка та налагодження програм або їх адаптація. Адаптація відбувається тоді, коли для створення інформаційної системи використовуються вже готові програми: типові або ті, які розроблялися для інших об'єктів. На кожен програму розробляється її опис або паспорт. Якщо програми адаптувались, то можуть бути описані тільки зміни, які були внесені до програм.

Сьомий етап. На етапі введення в експлуатацію необхідно виконати такий обсяг робіт: підготувати об'єкт експлуатацію, скомплектувати АІС, встановлюючи технічні та програмні засоби, виконати будівельно-монтажні роботи, провести попередні випробування системи, виконати дослідну експлуатацію системи та провести приймання в експлуатацію. Підготовка об'єкта до автоматизації починається з видання наказу про зміни в структурі об'єкта, документообігу, розподілі обов'язків між персоналом, переході на нову технологію обробки інформації. Такий наказ видається в довільній формі, але в ньому обов'язково наводиться термін переходу до нової технології та особи, які відповідають за впровадження й експлуатацію інформаційної системи. Для підготовки об'єкта

можуть розмножувати різноманітні посадові інструкції, бланки нових документів, готуватись класифікатори тощо. На цьому етапі дуже важливо підготувати персонал до роботи в інформаційній системі. Підготовка персоналу може провадитись силами розробників системи (лекції, семінари, практичні заняття) або з допомогою спеціальних курсів чи факультетів підвищення кваліфікації. Під час такого навчання кожний працівник має не тільки опанувати зміни у своїх посадових обов'язках, а й навчитися роботі з обчислювальною технікою. Таке навчання може передбачати засвоєння типових, найпоширеніших пакетів програмних засобів.

Паралельно з підготовкою персоналу провадяться роботи з установаження технічних та програмних засобів. Визначаються місця встановлення ЕОМ, засоби їх охорони, особи, відповідальні за супроводження системного програмного забезпечення, інсталюються необхідні пакети програм. У разі потреби виконуються будівельно-монтажні роботи, пов'язані з прокладанням кабелів, встановленням унікального обладнання, зміною освітлення місць, де встановлюються ЕОМ.

Попередні випробування системи виконує розробник, щоб перевірити коректність роботи технічних та програмних засобів, можливість використання прикладного програмного забезпечення.

Під час дослідної експлуатації заповнюють інформаційну базу на машинних носіях. Це роблять спеціалісти, які експлуатуватимуть інформаційну систему. На основі контрольного прикладу або реальних даних за конкретний період (період визначає користувач) виконуються основні розрахунки. За результатами дослідної експлуатації до програмного забезпечення можуть вноситись зміни. За домовленістю між користувачем і розробником системи може дороблятися й технічний проект.

Після завершення дослідної експлуатації відбуваються приймальні випробування, які можуть ґрунтуватись на аналізі документів, отриманих на ЕОМ, і порівнянні їх із документами, сформованими вручну. Випробування можуть провадитись спеціально створеною комісією, яка перевіряє роботу системи на

реальних або умовних даних у присутності членів комісії. Після приймальних випробувань, якщо робота інформаційної системи відповідає технічному завданню і реалізує всі передбачені функції, складається акт введення системи в експлуатацію.

Восьмий етап. Супроводження АІС виконується згідно з гарантійними зобов'язаннями розробника системи. У цей період можуть усуватися недоліки, які виявляються під час експлуатації.

Стадії та етапи, які мають бути пройдені під час створення АІС обумовлюються в договорах і технічному завданні .

1.3 Забезпечення функціонування та розвитку АІС на підприємстві

Зовнішнє середовище разом з підприємством як складовою частиною, утворює систему, що є інформаційним полем, в якому здійснюються різноманітні обміни інформацією. Певною мірою зовнішнє середовище здійснює керуючий вплив на підприємство з допомогою керуючих інформаційних продуктів. На поведінку підприємства можуть впливати також і інформаційні вироби - необов'язкова до виконання інформація. Вона може потрапляти на підприємство з точкових джерел інформації або з незалежних інформаційних нагромаджувачів .

Як керівна інформація так і некерівна може потрапляти до підрозділів підприємства та його персоналу різними шляхами: через органи управління підприємством , через інформаційно-комп'ютерний підрозділ (ІТ - підрозділ), через систему інформаційного обслуговування підприємства або безпосередньо до персоналу.

Управління інформаційними ресурсами зводиться до забезпечення їх своєчасного отримання та раціонального використання як системою управління

підприємством, так і всіма його підрозділами, а також своєчасного і максимально повного забезпечення інформацією зовнішніх споживачів.

Для здійснення управління необхідно [4] :

- знати інформаційні потреби системи управління підприємством і структурних підрозділів;
- мати регламент отримання інформації підрозділами;
- знати форму подачі необхідної інформації;
- знати джерела необхідної інформації;
- забезпечити регулярну взаємодію (прямий і зворотний зв'язок) системи управління та структурних підрозділів з джерелами інформації;
- забезпечити приймання, обробку, передачу користувачам та збереження інформації на підприємстві;
- здійснювати контроль та аналіз рівня та якості задоволення інформаційних потреб системи управління та структурних підрозділів;
- розробляти та реалізовувати управлінські дії, спрямовані на поліпшення інформаційного забезпечення підприємства;
- аналізувати ефективність функціонування інформаційної системи підприємства та впроваджувати заходи щодо його покращання;
- розробляти і реалізовувати стратегічні плани розвитку інформаційної системи підприємства та інформаційних технологій;
- налагоджувати зв'язки з зовнішніми організаціями-джерелами інформації на предмет підвищення якості інформаційного обміну.

Значна роль в управлінні інформаційними ресурсами на підприємстві відводиться його інформаційно-комп'ютерному підрозділу або, як його називають, підрозділу інформаційних технологій (ІТ - підрозділ). Такі підрозділи створюються і функціонують здебільшого на середніх і великих підприємствах.

Особливістю функціонування ІТ-підрозділу є те, що він окрім питань експлуатації інформаційних систем підприємства систематично займається їх розвитком. Річ в тому, що в процесі функціонування інформаційної системи

накопичується інформація про позитивні та негативні сторони ІС. Цьому сприяє ведення звітної документації за результатами експлуатації та її постійний аналіз. Крім цього, ІТ-підрозділ вивчає і аналізує потреби і вимоги ринку, світові тенденції щодо розвитку АІС, фінансові можливості підприємства, інвестиційний клімат на підприємстві, стан корпоративної культури.

Все це дозволяє ІТ-підрозділу розробляти і подавати на розгляд керівництву підприємства обґрунтовані пропозиції щодо стратегічного розвитку АІС.

ІТ-підрозділ є основним координатором всіх робіт з автоматизації управлінських процесів, з впровадження автоматизованих інформаційних систем на підприємстві, які здійснюється сторонніми організаціями. Він також забезпечує навчання кадрів-користувачів автоматизованих систем управління, здійснює консультації працівників підприємства щодо особливостей використання тих чи інших технічних засобів та програмних комплексів.

ІТ-підрозділ є головним консультаційним підрозділом керівництва підприємства щодо можливості чи необхідності впровадження на підприємстві інформаційних та автоматизованих систем управління. Керівник підрозділу повинен брати участь у всіх виробничих нарадах, що стосуються реорганізації або реструктуризації та розвитку підприємства, впровадження нових видів продукції, вдосконалення систем оплати праці, скорочення чисельності кадрів, укладання договорів на інформаційне обслуговування підприємства сторонніми організаціями, вибору організацій-виконавців розробки та впровадження автоматизованих систем управління та АІС.

Таким чином, успішне функціонування автоматизованих інформаційних систем на підприємстві буде залежати від фінансово-економічних, технічних і організаційних факторів підприємства. Якщо перші дві групи факторів залежать в основному від фінансових можливостей підприємства, то група організаційних факторів повністю підконтрольна системі управління підприємства.

Світова практика застосування сучасних методів реорганізації діяльності підприємств показала їх високу ефективність і універсальність. Особливо великі

можливості в цьому плані є у так званому «реінжинірингу» бізнес-процесів підприємства.

Координуюча роль щодо впровадження, експлуатації та розвитку інформаційних систем на підприємстві повинна належати спеціалізованому інформаційно-комп'ютерному підрозділу (ІТ-підрозділу), який повинен складатися з команди висококваліфікованих спеціалістів різного профілю, здатних постійно освоювати нову техніку і технології, навчатись самим і навчати цьому управлінський і виробничий персонал підприємства, гідно здійснювати місію провідника науково-технічного прогресу на підприємстві.

2. Огляд інформаційних систем для фондових ринків

Фондові ринки є дуже прибутковим і поширеним способом примноження капіталу. Як говориться в звіті Finance Magnates досить знати, що тільки на форексі щодня торгується більше 2 трильйонів доларів [6].

А якщо ми додамо до цієї цифри, що торгується на інших головних фондових ринках світу, таких, як Уолл-стрит, Лондон, Токіо та інші, ця цифра стане ошеломляючою, і фактично глобальні фондові ринки є фабрикою капіталу.

З колосальним технічним розвитком, включаючи алгоритми, штучний інтелект та машинне навчання, у всіх сферах життя, виникає питання, чи можна використовувати штучний інтелект для прогнозування руху ринків?

Відповідь, безумовно, так, використання штучного інтелекту для прогнозування руху фондового ринку в останній час виросло. Чи може сліпа довіра до цього методу привести до втрати капіталу трейдерів та інвесторів по всьому світу? У якому ступені ми можемо покладатися на цю технологію для прогнозування руху акцій з надійною точністю?

Сохраб Мохтарі, Кан Кі Інъ і Цзінь Лю, професори комп'ютерної інженерії Університету Флориди, обговорили це у своїй статті, озаглавленій «Ефективність штучного інтелекту в прогнозуванні фондового ринку на основі машинного навчання» [7].

Результати свідчать про те, що хоч штучний інтелект може передбачати тенденції цін на акції або загальні настрої щодо руху фінансових ринків, його точність недостатня. Понад те, хоча модель лінійної регресії може прогнозувати ціну з досить малою похибкою, вона не може точно передбачити те саме значення наступного робочого дня, тобто, ця здатність прогнозування дійсна лише протягом одного робочого дня.

Таким чином, ця модель, яка базується на штучному інтелекті, не може бути використана для довгострокових інвестицій. Точність алгоритмів прогнозування купівлі, продажу чи володіння акціями є недостатньо задовільною і може призвести

до втрати капіталу.

Грунтуючись на цих результатах, дослідники дійшли висновку, що штучний інтелект поки що не здатний передбачати рух фондового ринку з надійною та достовірною точністю.

2.1 Використання штучного інтелекту на фондових ринках

Застосування електронних технологій є однією з характерних рис розвитку сучасного фондового ринку: вони забезпечують високу динаміку операцій, істотно прискорюють розрахунки, розширюють коло учасників і зменшують ризики.

Фондова біржа – це організований, регулярно функціонуючий ринок цінних паперів. Фондові біржі покликані: мобілізувати тимчасово вільні кошти підприємств та населення у вигляді продажу ним цінних паперів, сприяти переміщенню грошового капіталу між різними економічними суб'єктами [8].

Штучний інтелект – це розділ інформатики, що включає розробку методів моделювання та відтворення за допомогою ЕОМ окремих функцій творчої діяльності людини, рішення проблеми представлення знань в ЕОМ та побудова баз знань, створення експертних систем, розробку, так званих інтелектуальних роботів [9].

Історія використання штучного інтелекту у роботі з інвестиційними проектами розпочалася у 1970-х роках в Америці. В наш час із розвитком потужності комп'ютерів розвивається і сфера їх використання: тепер часто машини самостійно приймають ті чи інші інвестиційні рішення у багатьох керуючих компаніях. Нейронна мережа, на технології якої заснована робота робота-співробітника керуючої компанії, розвивається на реально отриманих раніше історичних даних про роботу фондового ринку та про вже здійснені інвестиційні проекти. Кожна ситуація з портфельними інвестиціями математично являє собою звичайний набір змінних, саме так робот сприймає ті чи інші події та цифри. Перебираючи у своїй пам'яті схожі події, він знаходить ту модель поведінки, яка

максимально відповідає «правильному курсу» та реалізовує її. Таким чином, інвестиційні рішення, що приймаються машиною, складаються з аналізу реальних даних та досвіду попередників. Робот здатний обробити абсолютно всі індикатори ринку, інформація про які надійшла в систему, при цьому він може знайти взаємозв'язки між подіями, які людина навіть за дуже багатої фантазії пов'язати ніяк не може.

Недоліків штучного інтелекту на фондовій біржі практично не існує: він формує оптимальний на даний момент портфель, який приносить максимальну вигоду. Людський чинник у управлінні інвестиціями у своїй виключається, доходи підвищуються. Єдиний недолік машини - вона не може здогадатися, інтуїтивно приймаючи те чи інше нелогічне, але вдале рішення. Це прерогатива людини. При цьому людина ризикує, а машина прагне мінімізувати ризики.

Багато експертів абсолютно впевнені, що скоро «людська торгівля» зійде нанівець, бо вже зараз швидкість укладання угод перестає бути доступною людському розуму, не кажучи вже про руки. Це цілком може статися, чому підтвердження - дії деяких бірж, які не тільки запроваджують сертифікацію роботів і запроваджують плату за перевищення ліміту транзакцій, а й створюють спеціальні інструменти для цього.

Але серед професійних учасників ринку нерідко зустрічається скептичне ставлення до автоматичної торгівлі. Як правило, трейдерів бентежать три речі:

1. Людина не в змозі постійно контролювати правильність даних, що їх видає механічна торгова система або біржовий робот;
2. При перебої в системі, програма може довго час генерувати некоректні сигнали на здійснення угод, що призведе до збитків;
3. Багато гравців вважають, що інтернет-трейдинг повинен поєднувати в собі системний підхід і особисту інтуїцію, тобто суб'єктивну оцінку.

Загальносвітова тенденція дозволяє говорити про те, що торгових роботів на фондових майданчиках буде все більше. При цьому не виключено, що

довгострокова торгівля поступиться місцем короткостроковій, в рамках якої позиції відкриватимуться навіть не на секунди, а на мілісекунди. Не варто побоюватися, що фондові біржі перетворяться на великі зали з комп'ютерами без людей – роботи ефективні лише за торгівлі високоліквідними паперами, але безпорадні на ринках з невеликим оборотом коштів, адже за будь-якою «програмою роботом» стоїть все-таки людина .

2.2 Як використання штучного інтелекту на фондових ринках може сказатись на сфері фінансів і не тільки

У найближчому майбутньому більшу частину робочих місць на фінансових ринках займуть роботи, і це – хороша новина, тому що найкращі випускники університетів тепер зможуть піти у галузі з більш відчутною для населення та планети користю – технологічні стартапи, енергетику та медицину.

2.2.1 Штучний інтелект та рекордна прибутковість інвестицій

Більшість світових бірж використовують комп'ютери, що приймають рішення на основі алгоритмів та коректують стратегії з урахуванням нових даних, але деякі галузі, наприклад, ринки облігацій, автоматизуються повільніше. Нещодавно дослідницька група з університету Ерлангена-Нюрнберга у Німеччині розробила низку алгоритмів, які використали архівні дані ринків для тиражування інвестицій у режимі реального часу [10].

Одна з моделей дозволила отримати 73% повернення інвестицій щорічно з 1992 до 2015 року з урахуванням транзакційних витрат. Це можна порівняти з реальною ринковою прибутковістю в 9% на рік. Прибуток був особливо високим під час ринкових потрясінь 2000-го (545% прибутковості) та 2008-го років (681% прибутковості), що довело підвищену ефективність кількісних алгоритмів у періоди високої волатильності, коли на ринках переважають емоції.

Дослідження вчених в університеті Ерлангена-Нюрнберга показало, що у створених ними моделях прибуток від інвестицій знизився після 2001 року, оскільки використання роботів у торгівлі на біржі стало помітнішим і кількість можливостей для використання ринкової неефективності зменшилася. Проте в останні роки дохідність впала і іноді навіть ставала негативною, що дослідники пов'язують із зростаючим впливом роботів на біржову торгівлю .

Ідея використання комп'ютерів для торгівлі акціями не є новою. Її аналог – алгоритмічна торгівля – використовується вже понад десять років і неухильно набирає популярності. У 2012 році алгоритмічна торгівля займала 55% ринку [11].

Якщо цей тренд збережеться, 90% торгівлі вестиметься через комп'ютерні програми. Алгоритмічна торгівля сьогодні рухається у бік високочастотної HFT(High Frequency trading) - торгівлі, де акції купуються і продаються за частки секунди. Алгоритм швидко виявляє і використовує розбіжність, прибуток стає дедалі менше, але обсяг торгів не скорочується.

Ще одне дослідження EurekaHedge про 23 хедж-фонди, що використовують штучний інтелект, показало , що вони демонструють набагато кращі результати, ніж ті, що управляються людьми [12].

За останні шість років ці фонди досягли річної прибутковості в 8,44% порівняно зі звичайними фондами, показники яких становили від 1,62% до 2,62%. Автори дослідження пов'язують домінування штучного інтелекту у галузі про те, що він постійно проводить повторне тестування, а чи не просто накопичує дані. Це також може бути пов'язане з недоліками традиційних квантових підходів та застосуванням торгових моделей, побудованих з використанням неприбуткових тестів на історичних даних, які не здатні давати прибуток у режимі реального часу. Штучний інтелект нескінченно обробляє величезні масиви даних, включаючи книги, твіти, новини, фінансові показники та навіть розважальні телевізійні програми. Так він вчиться розуміти глобальні тренди і постійно вдосконалює свої прогнози фінансових ринків.

Хедж-фонди вже давно наймають на роботу математиків, які розробляють

статистичні моделі та використовують історичні дані для створення торгових алгоритмів, які передбачають можливості ринку, але штучний інтелект робить це швидше та постійно вдосконалюється.

Ось чому фінансові гіганти, такі як Goldman Sachs, який запусив торгову платформу Kensho на базі штучного інтелекту в 2014 році, переходять на роботизовані системи, що передбачають ринкові тренди і торгують значно краще за людей.

2.2.2 Чому штучний інтелект скоро витіснить людей із біржі

Заробити більше середнього на фондовому ринку майже неможливо – навіть найталановитіші інвестори на Уолл-Стріт не відрізняються постійністю. Трейдери та менеджери хедж-фондів не витримують конкуренції, проте їхня проблема полягає в тому, що вони просто люди, у той час як усі рішення, які приймають роботи, базуються лише на даних та статистиці [13]. «Люди завжди залишаються упередженими та емоційними, незалежно від того, усвідомлюють вони це чи ні, — в інтерв'ю Bloomberg говорив Бабак Ходжат, співзасновник фінансового стартапу Sentient та один із розробників Siri у Apple. — Усім відомо, що люди роблять помилки. На мою думку, набагато страшніше покладатися на здогади та інтуїцію, а не на дані та статистику».

Системи, на зразок тієї, що розробляє компанія Sentient, може аналізувати величезні обсяги інформації, що включають ринкові дані, обсяги торгів, коливання цін, інтернет-заявки SEC(комісія по цінним паперам) для всіх компаній, дані соцмереж, новини та відео на YouTube. Мета — добитися того, щоб алгоритм склав оптимальний інвестиційний портфель на основі наявних знань і регулярно оптимізував його, виходячи з очікуваних нових даних за кожен місяць.

Кількість подібних проектів останніми роками значно зросла. За деякими оцінками, у фінансовій сфері кількість компаній, що працюють зі штучним інтелектом, досягає 1500.

Наприклад, фонд Medallion у Renaissance Technologies, який використовує

кількісні методи аналізу фондового ринку, може похвалитися одними з найкращих показників в інвестиційній історії. За 20 років фонд зміг повернути +35% у річному вираженні. Це означає, що якби ви вклали \$10 тис у 1997 році, сьогодні у вас на руках було б уже \$ 4,04 млн.

Aidyia Limited, керуючий активами в Гонконгу, запустили хедж-фонд, що повністю керується штучним інтелектом. Він може читати новини кількома мовами, аналізувати економічні дані, виявляти сумнівні шаблони, прогнозувати ринкові тенденції та після цього інвестувати.

Деякі компанії використовують штучний інтелект задля забезпечення прибутковості через алгоритмічну торгівлю. Фонд Sentinent Technologies, лише за кілька хвилин може зімітувати 1800 торгових днів, зіштовхуючи трильйони віртуальних трейдерів між собою.

Багато хедж-фондів у всьому світі вже давно використовують машинне навчання для алгоритмічної торгівлі, тому що це виключає будь-які прояви ірраціональних почуттів, таких як страх і жадібність. Інвестори хочуть, щоб штучний інтелект розповів їм, як заробляти гроші на фондовому ринку.

2.2.3 Як це працює на практиці: 7 компаній з штучним інтелектом у сфері інвестицій

Numerai проводить змагання серед творців торгових стратегій. Елементи кращих стратегій потім використовуються фондом у реальній торгівлі на біржі, які творці отримують винагороду.

Qplum використовує машинне навчання для створення робота-консультанта, який застосовує алгоритми штучного інтелекту для ухвалення інвестиційних рішень.

Компанія **Sentinent** також запустила кілька програм на своїй платформі штучного інтелекту. На розвиток одного з них, пов'язаного з алгоритмічним продажем, вдалося залучити \$135 млн. Фонд створив кілька трильйонів роботів-трейдерів, пізніше об'єднав їх і має намір виділити цей проект в окрему компанію.

Alpaca, компанія, заснована в 2013 році, залучила \$1 млн на розробку трейдингової платформи Capitalico, яка дозволяє будувати біржові алгоритми на основі технічного аналізу, що прогнозує коливання вартості акцій. Платформа розпізнає шаблони користувача як «оптимістичні» і «песимістичні» і на основі цього будує торгову стратегію.

Французький стартап **Walnut Algorithms** залучив \$446 тис, щоб поєднати машинне навчання з фінансовою експертизою та досягти абсолютного повернення інвестицій.

Aidyia, гонконгський хедж-фонд, що використовує «загальний штучний інтелект», що більш точно імітує людський мозок, у 2015 році запустив фонд довгих/коротких інвестицій, що торгує акціями США та здійснює всі біржові угоди без втручання людини.

Канадська компанія **BUZZ** збирає дані із соціальних мереж, інтерпретує ці дані, використовуючи штучний інтелект, а потім визначає, у яких акцій прибутковість зросте. На основі цього будується індекс інтересу в соціальних медіа та визначаються 75 найпопулярніших акцій. Система аналізує не тільки Twitter та Facebook, а й більше 50 тематичних блогів, сайтів та новинних сервісів. Після цього Buzz очищає дані та інтерпретує настрої користувачів, маркує всі огляди продукту в мережі як позитивні чи негативні.

Якщо машини «переможуть»: найкращі «технарі» зможуть принести більше користі «Повсюдне впровадження штучного інтелекту у фінансовій індустрії призведе до того, що трейдери з величезними зарплатами втратять робочі місця через роботів так само швидко, як і фабричні робітники. — каже Марк Міневич, засновник Going Global Ventures та старший науковий співробітник американської Ради з конкурентоспроможності США. — Вплив штучного інтелекту в індустрії поступово наростає, але незабаром він зовсім змінить її».

Згідно з дослідженнями компанії Coalition Development, сьогодні середня зарплата співробітників у 12 найбільших інвестиційних банках сягає \$500 тис на рік, причому у багатьох трейдерів доходи дорівнюють кільком

мільйонам. Наприклад, у 2015 році як мінімум п'ять менеджерів топових хедж-фондів заробили \$1 млрд. Мотивація відмовитися від співробітників, які заробляють по \$500 на годину, та замінити їх роботами, зрозуміла.

У 2000 році Goldman Sachs, всесвітньо відомий банк, мав 600 трейдерів, які купували і продавали акції за вказівкою великих клієнтів банку, сьогодні залишилося лише два таких співробітники, а всю решту роботи роблять роботи. Коли те ж саме станеться з іншими фінансовими компаніями, питання часу.

Така тенденція, швидше за все, змінить індустрію, тому що найкращі випускники університетів втратять інтерес до Уолл-Стріт і віддадуть перевагу роботі в медицині, енергетиці, виробництві та інших корисних для суспільства сферах. Сьогодні приблизно третина випускників десяти найкращих бізнес-шкіл США йде працювати у фінанси, лише 5% йде в медицину і ще менша кількість – у всі інші галузі.

"Всіх цих розумних людей могли б найняти технологічні стартапи, у тому числі платформи, що розвивають штучний інтелект", - додає Міневич . Справді, коли хедж-фонди втратять інтерес до вчених та інженерів, вони зможуть приєднатися до технологічних стартапів, щоб розробляти інформаційні системи, проектувати безпілотні автомобілі, розвивати енергетичні технології, моделювати кліматичні зміни, ловити терористів та шукати ліки від раку, тобто займатися речами, корисними для широкого загалу.

3. Нейроні мережі та алгоритми прогнозування ціни

Оскільки основне завдання штучного інтелекту – моделювання міркувань, а природний «пристрій, здатний думати» – це мозок, очевидним завданням є створення «штучного мозку» за образом людського.

Нейрони – спеціалізовані клітини, здатні приймати, обробляти, кодувати, передавати та зберігати інформацію, організовувати реакції на подразнення, встановлювати контакти з іншими нейронами, клітинами органів. Унікальною особливістю нейрона є здатність генерувати електричні розряди та передавати інформацію з допомогою спеціалізованих закінчень – синапсів.

Число нейронів мозку людини наближається до 1000. На одному нейроні може бути до 10000 синапсів. Якщо тільки ці елементи рахувати осередками зберігання інформації, можна дійти висновку, що нервова система може зберігати 10^{19} одиниць інформації, тобто здатна вмістити майже всі знання, накопичені людством [14].

1943 став роком народження теорії штучних нейронних мереж. Маккалок та Пітт запропонували модель формального нейрона (Рис. 3.1) та описали основні принципи побудови нейронних мереж [15].

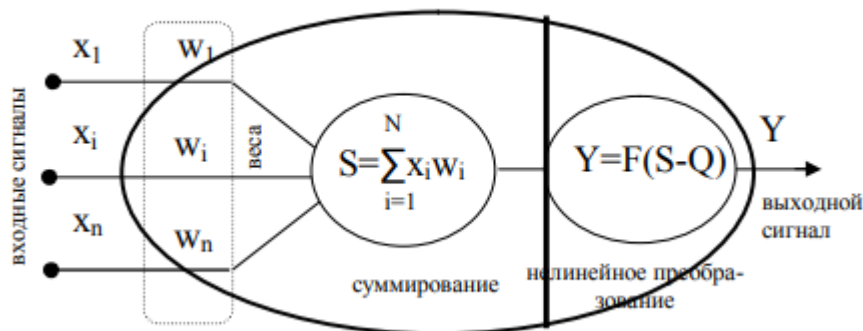


Рисунок 3.1 - модель формального нейрона.

3.1 Класифікація штучних нейронних мереж

За топологією (Рис 3.2, 3.3):

- Повнозв'язні (кожний нейрон пов'язаний з усіма іншими нейронами, у тому числі й сам із собою);
- Багатошарові (нейрони розташовуються шарами, і кожен нейрон наступного шару пов'язаний із усіма нейронами поточного шару).

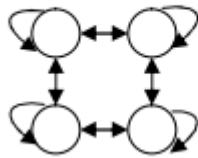


Рисунок 3.2 - Повнозв'язна
Нейрона мережа.

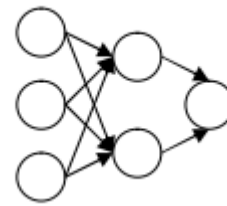


Рисунок 3.3 - Багатошарова
Нейрона мережа.

За організацією навчання:

- з вчителем (нейронну мережу навчають, подаючи на вхід значення навчальної вибірки та надаючи необхідні вихідні значення);
- без вчителя (на входи нейронної мережі подають безліч об'єктів, і нейронна мережа сама ділить їх на кластери чи класи).

За типами структур (Рис 3.4):

- нейрони з одним типом функції активації (усі нейрони мережі мають одну функцію активації $f(x)$, наприклад, лінійну);
- нейрони з кількома типами функцій активації (нейрони мережі мають різні функції активації).

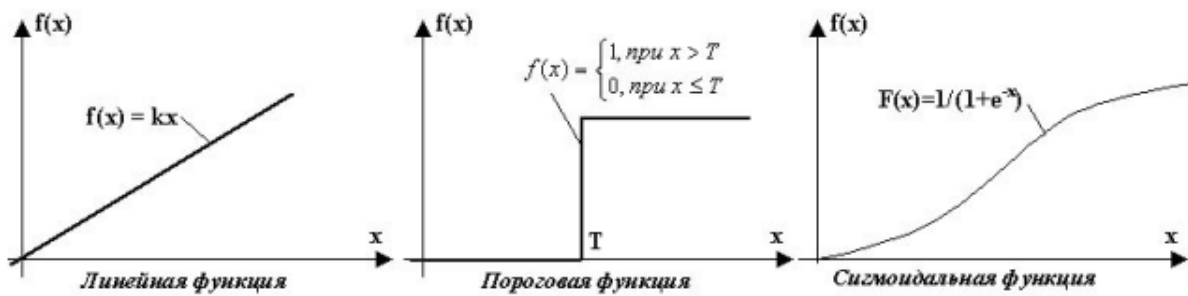


Рисунок 3.4 - Функції активації.

За типом зв'язків:

- прямого поширення (без зворотних зв'язків між нейронами; до таких мереж відносяться одношаровий та багатшаровий перцептрони, мережа радіальних базисних функцій);
- рекурентні (із зворотним зв'язком, від виходів нейронів до входів; до таких мереж відносяться змагальні мережі та мережа Хопфілда).

За типом сигналу:

- бінарні (на входи подаються лише нулі та одиниці);
- аналогові (на входи нейронів подаються значення безперервних функцій)

3.2 Одношарові штучні нейронні мережі

Один нейрон здатний виконувати найпростіші процедури розпізнавання, але лише з'єднання кількох нейронів здатне вирішити практично корисне завдання.

Найпростіша мережа (Рис. 3.5) складається із групи нейронів, що утворюють шар.

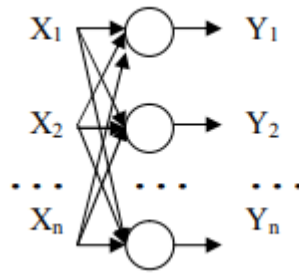


Рисунок 3.5 - Одношарова нейрона мережа.

3.3 Навчання по дельта-правилу

Дельта-правило є узагальненням алгоритму навчання персептрона. Дельта-правило працює тільки з безперервними функціями, що диференціюються, в режимі навчання з вчителем (supervised learning). Помилка, обчислювана у процесі навчання мережі – це функція, характеризує якість навчання цієї мережі, тому процес навчання нейронної мережі можна як процес мінімізації функції помилки. Напрямок зміни значення функції можна встановити, обчисливши похідну для функції одного змінного або градієнт для функції багатьох змінних. При мінімізації значення функції багатьох змінних менше значення необхідно шукати у бік антиградієнта.

У цьому алгоритмі навчання початкові ваги може бути будь-якими. Процес навчання можна вважати завершеним, якщо досягнута якась заздалегідь встановлена мінімальна помилка або алгоритм пропрацював обумовлену кількість разів.

Алгоритм навчання за дельта-правилом:

1 крок: ініціалізація матриці вагів (і порогів, у разі використання порогової функції активації) випадковим чином.

2 крок: пред'явлення нейронної мережі образу (на вхід подаються значення навчальної вибірки – вектор X , береться відповідний вихід – вектор D).

3 крок: визначення вихідних значень нейронної мережі (вектор Y).

4 крок: обчислення кожного нейрона величини розбіжності реального результату з бажаним:

$$\mathbf{e}_i = \mathbf{d}_i - \mathbf{y}_i, \quad (3.1)$$

де \mathbf{d}_i – бажане вихідне значення на i -нейроні, \mathbf{y}_i – реальне значення на i -нейроні.

5 крок: зміна ваг (і порогів при використанні порогової функції) за формулами:

$$\mathbf{w}_{ij}(\mathbf{t} + 1) = \mathbf{w}_{ij}(\mathbf{t}) - \mathbf{r} \cdot \mathbf{e}_i \cdot \mathbf{x}_j, \quad (3.2)$$

$$v_i(\mathbf{t} + 1) = v_i(\mathbf{t}) - \mathbf{r} \cdot \mathbf{e}_i, \quad (3.3)$$

де \mathbf{t} – номер поточної ітерації циклу навчання, \mathbf{w}_{ij} – вага зв'язку j -входу з i нейроном, \mathbf{r} – коефіцієнт навчання (задається від 0 до 1), \mathbf{x}_j – вхідний значення, v_i - Порогове значення i -нейрону.

6 крок: перевірка умови продовження навчання (обчислення значення помилки та/або перевірка заданої кількості ітерацій). Якщо навчання не завершено, то 2 крок, інакше закінчуємо навчання.

3.4 Багатошарові нейронні мережі

Одношарові нейронні мережі мають свої обмеження, багатошарові мережі вільні від них. Вони володіють великими обчислювальними можливостями. Хоча створені мережі всіх конфігурацій, які тільки можна собі уявити, пошарова організація нейронів копіює шаруваті структури певних відділів мозку.

Багатошарові мережі утворюються каскадами шарів. Вихід одного шару є входом для наступного шару, така організація мережі утворює багатошарова нейронна мережа з прямим поширенням (Рис 3.6).

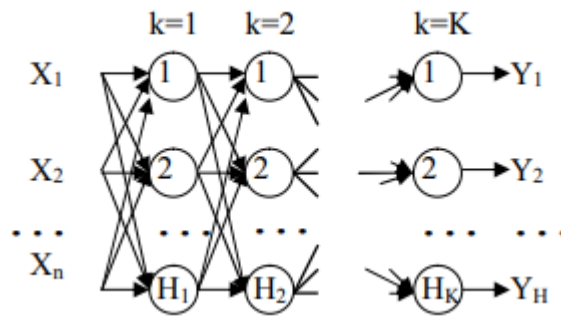


Рисунок 3.6 – Багатошарова нейронна мережа з прямим поширенням.

Багатошарова мережа може містити довільну кількість шарів (K), кожен шар складається з декількох нейронів, число яких також може бути довільно (N_k – кількість нейронів у шарі), кількість входів n , кількість виходів $N = N_K$ - числу нейронів у вихідному (останньому) шарі.

Шари між першим і останнім називаються проміжними або прихованими. Ваги у такій мережі мають три індекси: i – номер нейрона поточного шару, для якого зв'язок вхідний; j – номер входу або нейрона шару, для якого зв'язок вихідний; k – номер поточного шару в нейронній мережі (Для входів, вектора X , $k = 0$).

3.5 Навчання методом зворотного розповсюдження помилки

Навчання алгоритмом зворотного поширення помилки передбачає два проходи по всіх верствах мережі: прямий та зворотний.

При прямому проході вхідний вектор подається на вхідний шар нейронної мережі, після чого поширюється мережею від шару до шару. В результаті генерується набір вихідних сигналів, який є фактичною реакцією мережі на даний вхідний образ. Під час прямого проходу всі синаптичні ваги мережі фіксовані.

Під час зворотного проходу всі синаптичні ваги налаштовуються в відповідно до правила корекції помилок, а саме: фактичний вихід мережі віднімається з бажаного, внаслідок чого формується сигнал помилки. Цей сигнал згодом поширюється по мережі у напрямку, зворотному напрямку синаптичних зв'язків. Звідси і назва алгоритм зворотного розповсюдження помилки. Синаптичні

ваги налаштовуються з метою максимального наближення вихідного сигналу мережі до бажаного.

Алгоритм навчання за дельта-правилом:

1 крок: ініціалізація матриць вагів випадковим чином (у циклах).

2 крок: пред'явлення нейронної мережі образу (на вхід подаються значення навчальної вибірки – вектор X і береться відповідний вихід – вектор D).

3 крок (прямий прохід): обчислення в циклах виходів всіх шарів та отримання вихідних значень нейронної мережі (вектор Y):

$$y_i^k = f \left(\sum_{j=0}^{H_k-1} w_{ij}^k \cdot y_j^{k-1} \right), \quad (3.4)$$

$$y_j^0 = x_j, \quad (3.5)$$

$$y_0^{k-1} = 1, \quad (3.6)$$

$$x_0 = 1, \quad (3.7)$$

де y_i^k – вихід i -нейрону k -шару, f – функція активації, w_{ij}^k – синаптичний зв'язок між j -нейроном шару $k-1$ та i -нейроном шару k , x_j – вхідне значення.

4 крок (зворотний прохід): зміна ваги в циклах за формулами:

$$w_{ij}^k \cdot (t + 1) = w_{ij}^k(t) + r \cdot b_i^k \cdot y_j^{k-1}, \quad (3.8)$$

$$b_i^k = (d_i - y_i) \cdot y_i \cdot (1 - y_i), \quad (3.9)$$

– для останнього (вихідного) шару,

$$b_i^k = y_i \cdot (1 - y_i) \cdot \sum_{i=1}^{H_{k+1}} b_i^{k+1} \cdot w_i^{k+1}, \quad (3.10)$$

– для проміжних шарів, де t – номер поточної ітерації циклу навчання (номер епохи), r – коефіцієнт навчання (задається від 0 до 1), y_i^k – вихід i -нейрону k -шару, w_{ij}^k – синаптичний зв'язок між j -нейроном шару $k-1$ та i -нейроном шару k , d_i – бажане вихідне значення на i -нейроні, y_i – реальне значення на i -нейроні вихідного шару.

5 крок: перевірка умови продовження навчання (обчислення значення помилки та/або перевірка заданої кількості ітерацій). Якщо навчання не завершено, то 2 крок, інакше закінчуємо навчання. Середньоквадратична помилка обчислюється так:

$$e = \frac{1}{Q} \cdot \sum_{q=1}^Q \sum_{i=1}^N (d_i - y_i)^2 \quad (3.11)$$

де Q – загальна кількість прикладів, N – кількість нейронів у вихідному шарі, d_i – бажане вихідне значення на i -нейроні, y_i – реальне значення на i -нейрон вихідного шару.

3.6 Завдання, які вирішуються нейронними мережами

1. Класифікація образів. Завдання полягає у визначенні належності вхідного образу (наприклад, мовного сигналу або рукописного символу), представленого вектором ознак, до однієї або кількох заздалегідь визначених класів. До відомих програм належать розпізнавання літер, розпізнавання мови, класифікація сигналу електрокардіограми, класифікація клітин крові

2. Кластеризація/категоризація. При розв'язанні задачі кластеризації навчальна множина не має міток класів. Алгоритм кластеризації заснований на подібності образів і поєднує схожі образи в один кластер. Відомі випадки застосування

кластеризації для видобутку знань, стиснення даних та дослідження властивостей даних.

3. Апроксимація функцій. Припустимо, що є навчальна вибірка $((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n))$ (пари даних вхід-вихід), яка генерується невідомою функцією F , спотвореною шумом. Завдання апроксимації полягає у знаходженні невідомої функції F . Апроксимація функцій необхідна при вирішенні численних інженерних та наукових завдань моделювання.

4. Передбачення/прогноз. Нехай задані n дискретних відліків $\{y(t_1), y(t_2), \dots, y(t_n)\}$ у послідовні моменти часу t_1, t_2, \dots, t_n . Завдання полягає у передбаченні значення $y(t_{n+1})$ у наступний момент часу t_{n+1} . Передбачення/прогноз мають велике значення для прийняття рішень у бізнесі, науці та техніці (передбачення цін на фондовій біржі, прогноз погоди).

5. Оптимізація. Численні проблеми в математиці, статистиці, техніці, науці, медицині та економіці можуть розглядатися як проблеми оптимізації. Завданням алгоритму оптимізації є знаходження такого рішення, яке задовольняє системі обмежень та максимізує або мінімізує цільову функцію.

6. Пам'ять, адресована за змістом. У традиційних комп'ютерах звернення до пам'яті доступне лише за допомогою адреси, яка залежить від змісту пам'яті. Більше того, якщо припущено помилку у обчисленні адреси, то може бути знайдено зовсім іншу інформацію. Асоціативна пам'ять або пам'ять, адресована за змістом, доступна за вказівкою заданого змісту. Вміст пам'яті може бути викликаний навіть по частковому входу або за пошкодженого змісту. Асоціативна пам'ять може бути використана у мультимедійних інформаційних базах даних.

7. Управління. Розглянемо динамічну систему, задану сукупністю $\{u(t), y(t)\}$, де $u(t)$ – вхідний керуючий вплив, а $y(t)$ – вихід системи на момент часу t . У системах управління з еталонною моделлю метою управління є розрахунок такого вхідного впливу $u(t)$, при якому система діє бажаною траєкторією, заданою еталонною моделлю. Прикладом є оптимальне керування двигуном.

3.7 LSTM Модель

Long Short-Term Memory — це архітектура рекурентних нейронних мереж (RNN, штучна нейронна мережа), запропонована 1997 року Зеппом Хохрайтером та Юргеном Шмідгубером. Як і більшість RNN, мережа LSTM є універсальною в тому сенсі, що за достатньої кількості вузлів мережі вона може обчислювати будь-що, що може обчислювати звичайний комп'ютер, за умови, що вона має належну матрицю вагових коефіцієнтів, що може розглядатися як її програма. На відміну від традиційних RNN, мережа LSTM добре підходить для навчання з досвіду з метою класифікації, обробки або передбачення часових рядів в умовах, коли між важливими подіями існують часові затримки невідомої тривалості (Рис. 3.7) [16].

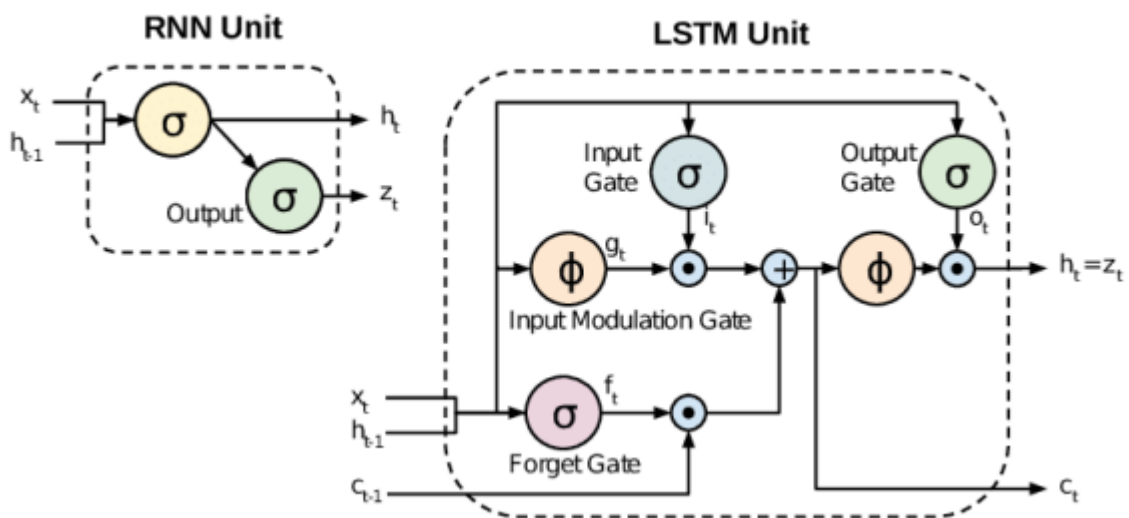


Рисунок 3.7 - Порівняння RNN моделі та LSTM моделі.

Вектор x і вектор h працюють як вхідні та вихідні вектори відповідно. Однак є ще кілька відмінностей.

По-перше, є ще два вхідних вектори : h_{t-1} і c_{t-1} . Ці два значення представляють відповідно короткостроковий і довгостроковий стан попереднього кроку. Таким чином, h_t є водночас вихідним сигналом поточної ітерації і вхідним короткостроковим сигналом для наступного кроку.

Основна ідея LSTM полягає в тому, що вона дозволяє довгостроковому вектору C текти з плином часу кроків, відкидати та вивчати нові дані (через Forget gate і Output Gate), в той же час також використовується короткострокова пам'ять h , яка використовується підрозділом.

3.7.1 Опис гіперпараметрів LSTM Моделі

У машинному навчанні гіперпараметр – це параметр, значення якого використовується для керування процесом навчання. На відміну від цього, значення інших параметрів (як правило, ваги вузлів) виводяться за допомогою навчання.

Гіперпараметри можна класифікувати як гіперпараметри моделі, які неможливо визначити під час підгонки машини до навчального набору, оскільки вони відносяться до задачі вибору моделі, або гіперпараметри алгоритму, які в принципі не впливають на продуктивність моделі, але впливають на швидкість і якість навчального процесу. Прикладом гіперпараметра моделі є топологія та розмір нейронної мережі. Прикладами гіперпараметрів алгоритму є швидкість навчання та розмір пакету даних. Розмір пакету може посилатися на повну вибірку даних, де розмір міні-партії буде меншим набором вибірки.

Мережа LSTM має кілька гіперпараметрів, які можна налаштувати. Найважливішими серед них є:

- Розмір пакета: визначає кількість зразків, які проходять через мережу перед самим оновленням внутрішніх параметрів мережі. Кожне оновлення внутрішніх параметрів називається ітерацією, тому певним чином визначення розміру пакету дорівнює кількості ітерацій у кожному епоху (курсі навчання).

Наприклад, із навчальним набором даних із 11 520 спостережень, розміром пакету в 96 вибірок дозволить нейронній мережі проробити 120 ітерацій.

$$\frac{11520}{96} = 120, \quad (3.12)$$

- Кількість епох: визначає кількість ітерацій для всього навчального набору даних.
- Кількість шарів: шари поділяються на вхідні, приховані та вихідні і є агрегаціями нейронів. Але як гіперпараметр кількість шарів відноситься лише до прихованих шарів.
- Одиниці на шар відноситься до кількості нейронів у кожному шарі.
- Довжина вхідної послідовності (також називається часовими кроками): кількість спостережень у минулому, які містить кожна вибірка.
- Об'єктивна функція (або функція втрат): функція, яка обчислює різницю між прогнозованим результатом і фактичним, тобто функція, яку нейронна мережа намагається мінімізувати. Для цього проекту, оскільки метою є мінімізувати помилку, вона буде називатися функцією втрат.
- Оптимізатор: функція оптимізації - це механізм, який використовує нейронну мережу для пошуку значення, яке мінімізує функцію втрат. Одним з найважливіших алгоритмів оптимізації є стохастичний градієнтний спуск (SGD), який спирається на ітераційне обчислення градієнта функції, роблячи кроки в кожній точці до негативного градієнта, щоб знайти глобальний мінімум. Однак існують новіші алгоритми, які розширюють і модифікують цей підхід, як-от AdaGrad або Adam, і які зближуються швидше і знаходять кращі мінімуми через свою точнішу поведінку в розріджених градієнтах, які з'являються, наприклад, при наближенні до мінімуму [17].

3.8 Модель Facebook Prophet

Facebook Prophet — це модель і бібліотека, які надають функції як

узагальнених лінійних моделей, так і адитивних моделей, в основному розширюючи лінійні моделі за допомогою нелінійних функцій згладжування. Це було визначено Тейлором і Летемом у 2017 році.

Основна відмінність **Prophet** від інших статистичних методів полягає в підході аналітика. Цей підхід дозволяє аналітику застосувати свої знання в області даних до алгоритму прогнозування, не маючи жодних знань про статистичні методи, що працюють зсередини. Таким чином, цей підхід намагається скористатися перевагами як статистичного прогнозування, так і прогнозування з оцінками, причому останнє є методами прогнозування, заснованими на рішеннях експертів. Параметри, які можуть бути налаштовані аналітиком, будуть визначені та пояснені нижче. Загальна функція для визначення часового ряду така:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t, \quad (3.13)$$

де $g(t)$ представляє неперіодичні зміни значення часового ряду, $s(t)$ сезонність моделі (яка може бути щоденною, щотижневою, щомісячною, щорічною або будь-якою іншою), $h(t)$ представляє наслідки свят і ϵ_t є помилкою [18].

3.8.1 Гіперпараметри у Facebook Prophet's

Існує кілька реалізацій параметрів для налаштування, основними з яких є:

- **Changepoints:** точки, які визначають зміни тенденцій. Їх можна знайти за допомогою самого алгоритму або їх також може визначити та налаштувати аналітик.
- **Сезонність,** яка визначає періодичні функції, які можуть впливати на часові ряди. За замовчуванням Prophet враховує річну, тижневу та щоденну сезонність і намагається знайти тенденції, які представляють ці періодичні ефекти в даних.
- **Свята:** особливі дні (свята або будь-яка інша повторювана подія) також

можна моделювати за допомогою адитивної моделі Prophet. Дати можна додати вручну, але є також вбудовані святкові набори для десятка країн.

З цих гіперпараметрів, «Свята» був єдиним параметром, який не був налаштований або врахований для побудови поточної моделі. Головною причиною для такого рішення є те, що ринок акцій закритий у святкові дні Сполучених Штатів, тому наш оригінальний набір даних не містить жодної інформації в ці дні для моделювання алгоритму.

4. Розробка інформаційної системи для прогнозування ціни акцій

4.1. Опис структури вхідних даних

Найчастіше набір даних містить вміст однієї таблиці з базою даних або статистичної матриці даних. Кожен стовпець таблиці представляє певну змінну, і кожен рядок відповідає даному елементу відповідного набору даних.

Для реалізації інформаційної системи було використано декілька баз даних, які відповідали компаніям для яких робиться прогноз, а саме SPY, QQQ, JNJ, NKE, MCD, AAPL, AMZN, NFLX, TSLA, GLD, USO. Всі дані було взято з публічного набору даних на сервісі Yahoo Finance [19].

Бази даних складаються з одного елемента – це csv файл, який містить 7 атрибутів. Для кожної компанії маємо свій файл з даними за кожен робочий день з 2015 року.

Опис атрибутів бази даних :

1. Date : дата , за який день дані у цій строчці.
2. Open : ціна відкриття акції на біржі США
3. High : максимальна вартість акції за день на біржі США
4. Low : мінімальна вартість акції за день на біржі США
5. Close : ціна закриття акції на біржі США
6. Adj Close : ціна закриття , враховуючи сплату дивідендів та спліти [20]
7. Volume : кількість акцій, які були куплені чи продані в поточний день.

Наведемо приклад вигляду бази даних на Рис. 4.1

	Date	Open	High	Low	Close	Adj Close	Volume
0	2015-01-02T00:00:00	206.3800	206.8800	204.1800	205.4300	179.7842	121465900
1	2015-01-05T00:00:00	204.1700	204.3700	201.3500	201.7200	176.5373	169632600
2	2015-01-06T00:00:00	202.0900	202.7200	198.8600	199.8200	174.8745	209151400
3	2015-01-07T00:00:00	201.4200	202.7200	200.8800	202.3100	177.0536	125346700
4	2015-01-08T00:00:00	204.0100	206.1600	203.9900	205.9000	180.1954	147217800
5	2015-01-09T00:00:00	206.4000	206.4200	203.5100	204.2500	178.7515	158567300
6	2015-01-12T00:00:00	204.4100	204.6000	201.9200	202.6500	177.3512	144396100
7	2015-01-13T00:00:00	204.1200	205.4800	200.5100	202.0800	176.8523	214553300
8	2015-01-14T00:00:00	199.6500	201.1000	198.5700	200.8600	175.7846	192991100
9	2015-01-15T00:00:00	201.6300	202.0100	198.8800	199.0200	174.1743	176613900
10	2015-01-16T00:00:00	198.7700	201.8200	198.5500	201.6300	176.4586	211879600
11	2015-01-20T00:00:00	202.4000	202.7200	200.1700	202.0600	176.8348	130991100
12	2015-01-21T00:00:00	201.5000	203.6600	200.9400	203.0800	177.7275	122942700
13	2015-01-22T00:00:00	203.9900	206.2600	202.3300	206.1000	180.3705	174356000

Рисунок 4.1 – Приклад вигляду бази даних.

4.2 Основні принципи обробки даних для інформаційної системи

Одним з ключових питань при використанні нейронних мереж завжди є питання про поділ і обробку вхідних даних. Питання обробки в даному випадку не є актуальним оскільки ми маємо повноту даних для нашої задачі .

Поділ датасету був класичним , 70 % пішло на навчання , а залишок (30 %) на тестування. Хоч і здається що даних за 5 років (2015-2020) має бути достатньо для гарного навчання нейронної мережі , але якщо порахувати в кількості записів в базі даних то отримаємо всього близько 1000 записів, що звісно є не достатнім для сучасних нейронних мереж.

4.3 Генерація прогнозу та різниця між запропонованими моделями

Модель 1 - LSTM модель здатна робити прогноз на 1 один день опираючись на 60 днів до цього. Оскільки у нас вхідні дані одразу без вихідних, то це приблизно

3 місяці для передбачення одного дня. Ця модель добре проявляє себе, тобто має дуже високу вірогідність близького передбачення ціни. Ще її можна назвати тренд передбаченням, тому що ця модель гарно розуміє тенденцію поведінки ціни. Звісно ця модель зовсім не може передбачати чогось неочікуваного, так званого «Чорного лебедя» [21].

Модель 2 – FbProphet модель здатна робити прогноз більше ніж один день, навіть на 5 років, якщо користувач цього захоче. Ця модель опирається на всі 70 % тренувальних даних для прогнозу. Звісно, порівняно з LSTM моделлю, вірогідність не така висока, до того ж ця модель передбачає діапазон цін а не конкретну ціну. Чим далі у майбутнє запит, тим ширше ми маємо діапазон. Ця модель також зовсім не підходить для передбачення краху ринку.

Тож, маємо дві моделі, з трохи різним функціоналом передбачення. Саме через це їх можна використовувати як в купі для передбачення та аналітики, так і окремо. Звісно ці моделі не можна відокремлювати від реальних даних на довгий час, тобто системи з динамічним оновленням та поповненням даних більше всього підходять для цих моделей.

4.4 Основні принципи проведення тестів

Використовуючи обидві моделі було розроблено 4 стратегії і проведено тести на тренувальних даних (2020 – 2022 р.). Після опису стратегій наведено матрицю дохідності для кожної стратегії і компанії (Рис. 4.2).

1. Стратегія полягає в тому, що якщо LSTM модель передбачає збільшення ціни акцій, то ми покупаємо і продаємо по прогнозу якщо до нього дійшло або по ціні закриття акції під кінець торгової сесії.

Стратегія гарно працює на компаніях з великою капіталізацією.

Сумарний процент прибутковості для нашого списку компаній : 77,69 % за два роки.

2. Наступна стратегія – це доповнення першої стратегії. Полягає в тому, що ми

пропускаємо наш сигнал на покупку, якщо прогнозований зріст менше 1 %. Також гарно працює на компаніях з великою капіталізацією. Сумарний процент прибутковості для нашого списку компаній : 145,84% за два роки.

3. Третя стратегія також виступає доповненням першої і одночасно другої. Полягає в тому, що якщо акції компанії впали на 1% за день і LSTM модель прогнозує збільшення більше ніж на 1%, то ми купуємо і продаємо по прогнозу або по ціні закриття. Сумарний процент прибутковості для нашого списку компаній : 286,51% за два роки.
4. Четверта стратегія використовує модель FbProphet і націлена на компанії які мають тенденцію до росту. Суть стратегії полягає в тому щоб «відкупити» велике падіння якщо ми маємо прогноз на збільшення цін через місяць. Таким чином кожне падіння 5% та більше ми купуємо і продаємо по прогнозу або ж через місяць, якщо до ціни прогнозу не доходить . Сумарний процент прибутковості для нашого списку компаній : 610,72 % за два роки.

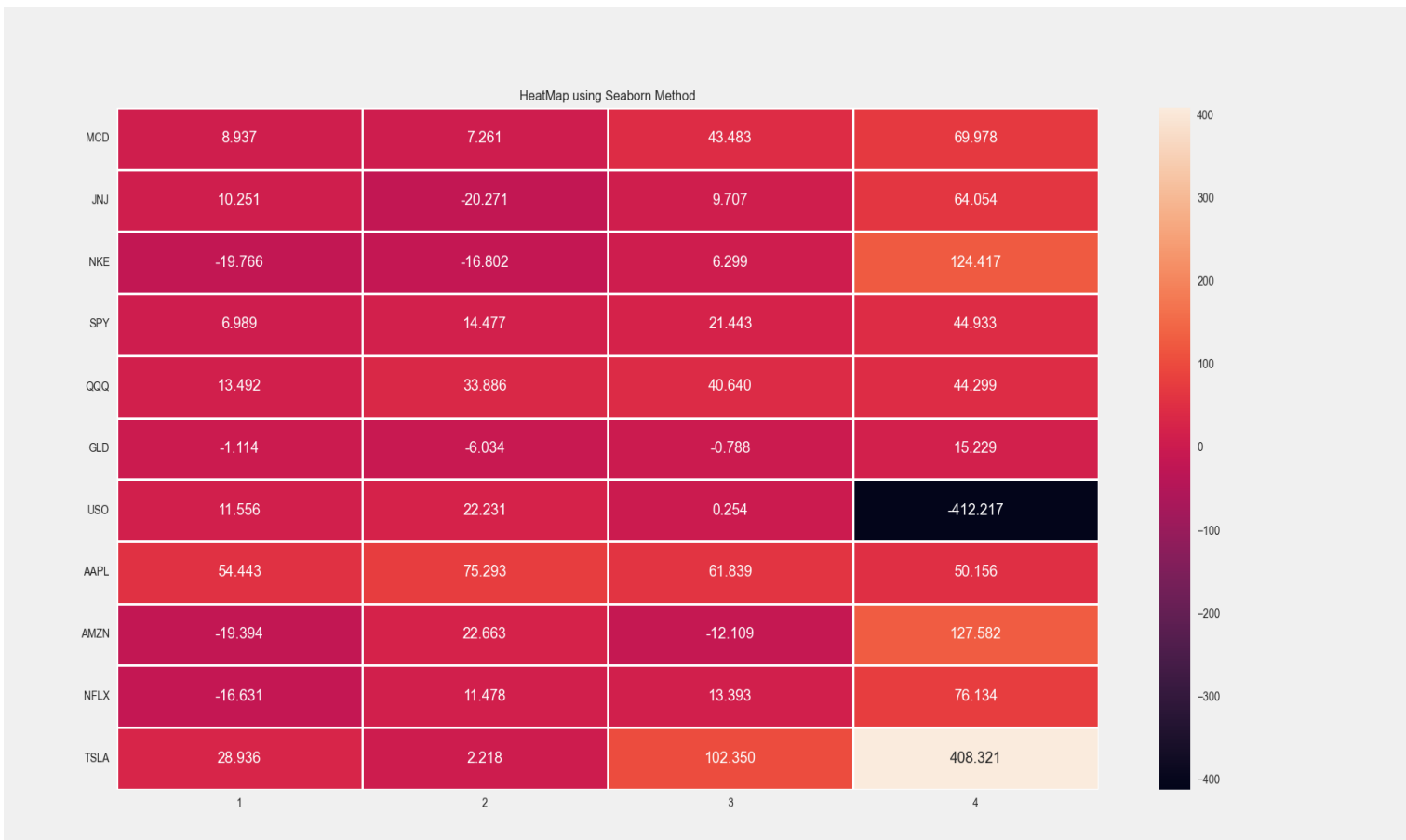


Рисунок 4.2 - Матриця дохідності стратегій.

Обов'язково треба додати, що такі акції як TSLA, AAPL, AMZN, NFLX зробили дуже потужний ріст у ціні саме у період з 2020 – 2022 , коли більшість компаній відновлювалися після кризи COVID 19 , тому це певним чином підвищило відсоток прибутковості наших тестів. Для повного розуміння ситуації, акції компанії Тесла(TSLA) вирости з травня 2020 по травень 2022 року приблизно на 700%, в той час як акції найбільшої технологічної компанії Apple(AAPL) в той самий період вирости майже на 150%. Однак були й такі компанії, як USO – нафта США , які потерпіли крах і вперше в історії фондових ринків ми бачили від'ємну ціну на ф'ючерс нафти. Саме через те, що ми дотримувались принципу диверсифікації, тобто рівномірного розподілу між різними та некорельованими цінними паперами, в решті решт залишились в прибутку , навіть коли мали найгірший результат для 4 стратегії для компанії USO в -412 % прибутку.

4.5 Інтерфейс користувача

Для забезпечення ефективної роботи користувача з рекомендаційною системою, було створено інтерактивний веб-портал.

Серед багатьох підходів до імплементації функціональних веб-порталів, було обрано комбінацію мови Python та бібліотеки Streamlit, яка поєднує в собі можливості для створення клієнтської та серверної частини (front-end and back-end).

Після відкриття веб-порталу, користувач має обрати компанію зі списку для роботи. Після цього він отримує необроблені дані з бази даних для цієї компанії. Також користувач отримує графік цін компанії з 2015 року (Рис. 4.3).

Користувач має можливість зробити прогноз на один день, чи на декілька днів чи навіть на декілька років, двома різними моделями, для цього йому дається повзунок, яким він обирає період на скільки зробити прогноз. Після того як нейрона мережа обробила запит та зробила прогноз, користувач отримує інтерактивний графік з прогнозом. Для прикладу наведено прогноз на 1 рік для акції SPY (Рис. 4.4).

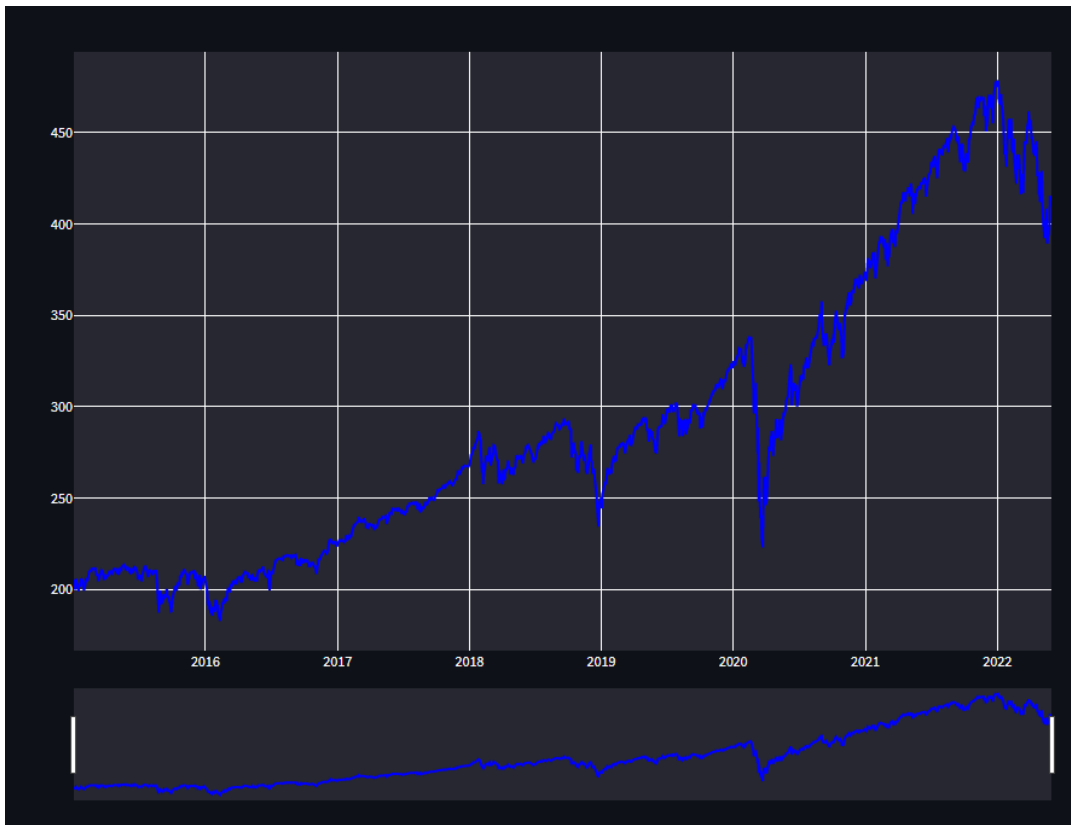


Рисунок 4.3 – Графік ціни компанії.

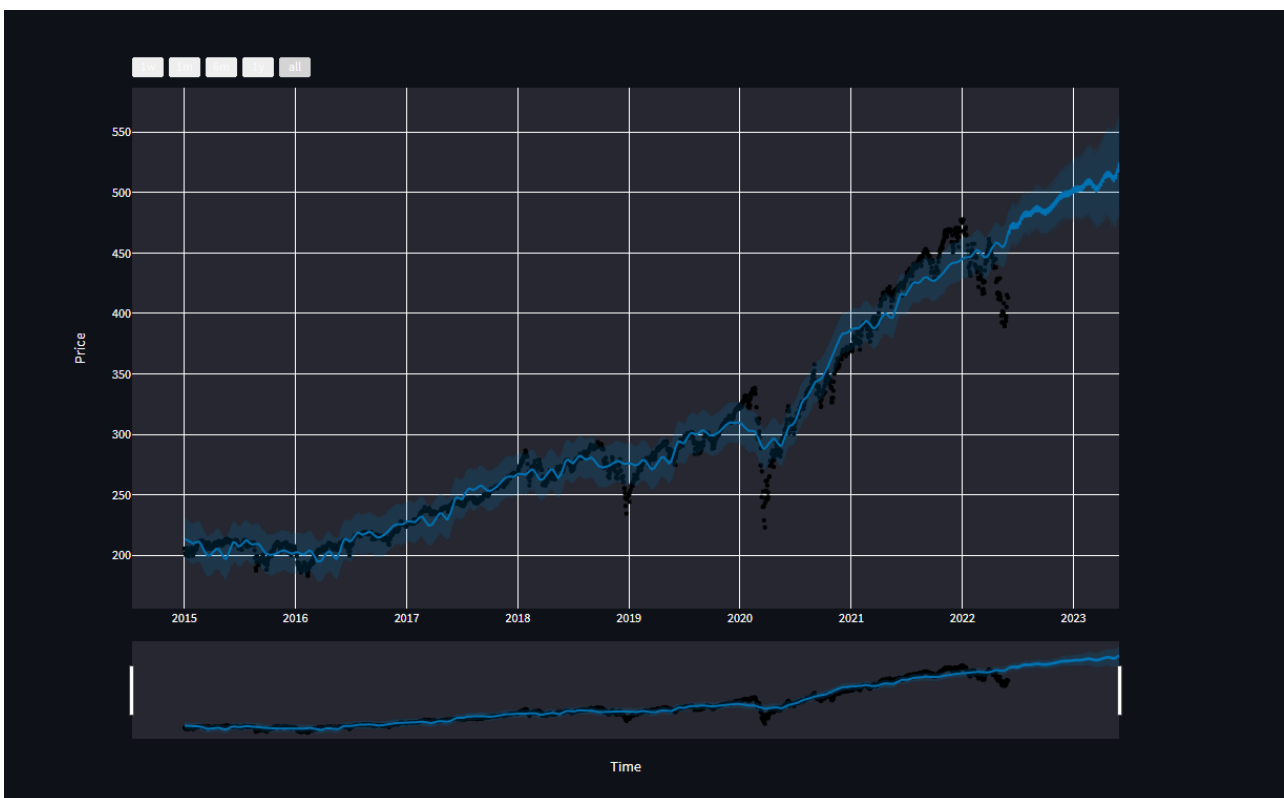


Рисунок 4.4 – Прогноз на 1 рік для SPY.

Також користувач отримує результати тестів які були розписані вище, та інтерактивний графік зміни дохідності і ціни акцій (Рис. 4.5).



Рисунок 4.5 – Графік прибутку разом з графіком ціни.

Висновки

В кваліфікаційній роботі було досліджено основні принципи побудови та види інформаційних систем, а також труднощі які виникають під час їх розробки, серед яких недостатня кількість інформації (тренування нейронних мереж потребує значної кількості добре-розподілених даних). Для розв'язання практичної задачі було запропоновано інформаційну систему побудовану на основі двох нейронних моделей LSTM та FbProphet. Запропонована система використовує декілька публічних датасетів з ринку цінних паперів США, проте це не єдиний ринок на якому дана система може бути використана. Датасет що досліджувався, містить в собі набір даних про коливання (відкриття, закриття, максимум, мінімум) цін конкретної акції за кожен день. На основі цих даних було застосовано модель LSTM для прогнозу наступного дня і модель FbProphet для прогнозування майбутнього періоду аж до 5 років. Опираючись на вихідні результати обох моделей було проведено ряд тестів на даних з 2020 по 2022 роки для оцінки прибутковості. Для зручності використання рекомендаційної системи, було створено інтерактивний веб-портал в який було інтегровано реалізовану систему.

Запропонована інформаційна система виявилась досить ефективною, прогнози були візуально схожі на реальні дані та ряд тестів показав дуже гарну прибутковість. Проте, система має недоліки пов'язані з високою залежністю від реальних даних, невмінням прогнозувати неочікувані коливання акцій. Можливо, якщо поєднати прогнозування нейронних моделей з технічним аналізом та аналізом соціальних мереж для конкретної акції то ці недоліки вдасться виправити.

Список використаних джерел

1. *История борьбы: гроссмейстер против компьютера. Когда в шахматах компьютер стал недостижим для человека?* - Геннадий Козлов [Электронный ресурс] // BanksToday – Режим доступа до ресурсу: <https://bankstoday.net/last-articles/istoriya-borby-grossmejster-protiv-kompyutera-kogda-v-shahmatah-kompyuter-stal-nedostyagaem-dlya-cheloveka>.
2. Гужва В. М.. «Інформаційні системи і технології на підприємствах», Київ, КНЕУ, 2001, сс. 7 – 30.
3. Скороход О.М.. «Конспект лекцій з дисципліни Інформаційні системи і технології на підприємствах», Херсон, 2009, сс 6-35.
4. Jannach, D., Zanker, M., Felfernig, A., Friedrich, G.: *Recommender Systems An Introduction*. Cambridge University Press (2010), pp.14– 20
5. Ricardo Baeza– Yates, Berthier Ribeiro– Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.
6. Bilal Jafar (2022, April). *Tradeweb Reports ADV of \$1.09 Trillion in April 2022*- [Электронний ресурс] // FinanceMagnates– Режим доступа до ресурсу: <https://www.financemagnates.com/institutional-forex/tradeweb-reports-adv-of-109-trillion-in-april-2022/> .
7. Sohrab Mokhtari, Kang K. Yen, Jin Liu. *Effectiveness of Artificial Intelligence in Stock Market Prediction based on Machine Learning*. International Journal of Computer Applications, 2018.
8. Eugene F Fama. “Efficient capital markets”. *The journal of finance* 46.5 (1991), pp. 1575–1576.
9. Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2017, p.17.
10. Casal Sandro; Ploner Matteo; Sproten Alec N. *Fostering the best execution regime: An experiment about pecuniary sanctions and accountability in fiduciary*

money management. IWQW Discussion Papers, No. 13/2014

11. Clara Furse, Philip Bond, Dave Cliff, Charles Goodhart. The Future of Computer Trading in Financial Markets An International Perspective . The Government Office for Science, 2012, pp.12-14.
12. May figures from Eureka hedge show hedge funds outperforming NASDAQ and S&P 500 - [Электронный ресурс] // InstitutionalAssetManager– Режим доступа до ресурсу : <https://www.institutionalassetmanager.co.uk/2022/05/24/309927/may-figures-eureka-hedge-show-hedge-funds-outperforming-nasdaq-and-sp-500>
13. Partha S Mohanram. “Separating winners from losers among low book-to-market stocks using financial statement analysis”. Review of accounting studies 10.2-3 (2005), pp. 153–170.
14. Dr Alan Woodruff. *What is a neuron?*. [Электронный ресурс] // The University of Queensland, Queensland Brain Institute – Режим доступа до ресурсу: <https://qbi.uq.edu.au/brain/brain-anatomy/what-neuron>
15. Ясницкий Л. Н. Введение в искусственный интеллект : учебное пособие для вузов по математическим направлениям и специальностям , 2010, сс. 2-32
16. Alejandro Mata. A comparison between LSTM and Facebook Prophet models: a financial forecasting case study. Gener, 2020. Pp. 15-19.
17. Derya Soydaner. “A Comparison of Optimization Algorithms for Deep Learning”. February 2020, International Journal of Pattern Recognition and Artificial Intelligence.
18. Oskar Triebe, , Hansika Hewamalagec, Polina Pilyuginad, Nikolay Laptev, Christoph Bergmeirc, Ram Rajagopala. “Neural Prophet: Explainable Forecasting at Scale”. [Электронный ресурс] // Stanford University – Режим доступа до ресурсу: <https://arxiv.org/pdf/2111.15397.pdf>
19. Stock-Market Sentiment Dataset. [Электронный ресурс] // Yahoo Finance – Режим доступа до ресурсу:

<https://finance.yahoo.com/quote/YHOO/history?p=YHOO>

20. A. Ganti, G. Scott.(December, 2020). *Adjusted Closing Price*. Электронный ресурс] // Investopedia– Режим доступа до ресурсу:
https://www.investopedia.com/terms/a/adjusted_closing_price.asp
- 21.Нассим Николас Талеб. “Черный лебедь. Под знаком непредсказуемости”.
2017.

Додаток А. Код програмної реалізації

```
import datetime
import math
import warnings
from datetime import date

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import pandas_datareader as web
import streamlit as st
import yfinance as yf
from keras.models import load_model
from PIL import Image
from plotly import graph_objs as go
from prophet import Prophet
from prophet.plot import plot_plotly
from sklearn.preprocessing import MinMaxScaler

warnings.filterwarnings("ignore")

plt.style.use('fivethirtyeight')

START = "2015-01-01"
TODAY = date.today().strftime("%Y-%m-%d")

st.title("Оцінка вартості та прогноз ціни акцій фондового ринку на  
прикладі ринку цінних паперів США")

stocks = ("SPY", "QQQ", "JNJ", "NKE", "MCD", "AAPL", "AMZN",
          "NFLX", "TSLA", "GLD", "USO")
selected_stocks = st.selectbox("Оберіть компанію для оцінки та прогнозу :",
                               stocks)

my_width = 1100
my_height = 800
```



```

@st.cache(suppress_st_warning=True)
def load_data(ticker):
    data = yf.download(ticker, START, TODAY)
    data.reset_index(inplace=True)
    return data

data_load_state = st.text("Отримуємо дані для цієї компанії ... ")
data = load_data(selected_stocks)
data_for_strategy = yf.download(selected_stocks, START, TODAY)
data_load_state.text("Дані отримано! ")

st.subheader('Необроблений датасет')
st.write(data)

@st.cache(suppress_st_warning=True)
def plot_raw_data():
    fig = go.Figure()
    fig.add_trace(go.Scatter(x=data["Date"], y=data["Close"], name='Stock
Close'))
    fig.layout.update(xaxis_rangeslider_visible=True)
    fig.update_layout(width=my_width, height=my_height)
    st.plotly_chart(fig, width=my_width, height=my_height)

st.subheader('Покажемо на графіку історію цін акцій компанії')
plot_raw_data()

@st.cache(suppress_st_warning=True)
def make_prediction(ticker, parametr, for_one_day,
date_for_one_day=TODAY):
    # get the stock
    df = web.DataReader(ticker, data_source='yahoo', start=START,
end=TODAY)
    # df = data
    # create new dataframe with only the 'close' column
    data_for_pred = df.filter([parametr])
    dataset = data_for_pred.values

```

```
training_data_len = math.ceil(len(dataset) * 0.7) # put 70% of our data to
train and 30 % to test with
```

```
# scale the data
```

```
scaler = MinMaxScaler(feature_range=(0, 1))
```

```
scaled_data = scaler.fit_transform(dataset)
```

```
days_to_train = 60
```

```
# Build the LSTM model
```

```
model = load_model('spy_closes_5epochs.h5')
```

```
if not for_one_day:
```

```
    # create test dataset
```

```
    test_data = scaled_data[training_data_len - days_to_train:, :]
```

```
    x_test = []
```

```
    for i in range(days_to_train, len(test_data)):
```

```
        x_test.append(test_data[i - days_to_train:i, 0])
```

```
x_test = np.array(x_test)
```

```
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))
```

```
# get the models predicted price values
```

```
predictions = model.predict(x_test)
```

```
predictions = scaler.inverse_transform(predictions)
```

```
train = data_for_pred[:training_data_len]
```

```
valid = data_for_pred[training_data_len:]
```

```
valid['Predictions'] = predictions
```

```
return valid, train # returns data: Date(index), Close, Predictions
```

```
else:
```

```
    if datetime.datetime.strptime(TODAY, "%Y-%m-%d").date() <
date_for_one_day:
```

```
        # st.write(datetime.datetime.strptime(TODAY, "%Y-%m-%d").date())
```

```
        # st.write(date_for_one_day)
```

```
        # st.write(datetime.datetime.strptime(TODAY, "%Y-%m-%d").date()
< date_for_one_day)
```

```
        return "Модель на основі LSTM не вміє робити прогнози більше ніж
```

```

на 1 день у майбутнє\nЯкщо ви обрали завтрашню дату і " \
    "отримали це повідомлення , то проблема в тому, що ринки ще
відкриті і не має даних за сьогодні! " \
    "Також через це ви не побачили реальних даних на цю дату "
else:
    quote = web.DataReader(ticker, data_source='yahoo', start=START,
end=date_for_one_day)
    parametr1 = "Low"
    parametr2 = "High"
    new_df = quote.filter([parametr])
    new_df2 = quote.filter([parametr2])
    if datetime.datetime.strptime(TODAY, "%Y-%m-%d").date() !=
date_for_one_day:
        st.write("Real Data", " Low-High = ", float(new_df[parametr1][-1]),
'- ',
                float(new_df2[parametr2][-1]))
    last_60_days = new_df[-60:].values
    last_60_days_scaled = scaler.transform(last_60_days)
    X_test = []
    X_test.append(last_60_days_scaled)
    X_test = np.array(X_test)
    X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
    pred_price = model.predict(X_test)
    pred_price = scaler.inverse_transform(pred_price)
    last_60_days2 = new_df2[-60:].values
    last_60_days_scaled2 = scaler.transform(last_60_days2)
    X_test2 = []
    X_test2.append(last_60_days_scaled2)
    X_test2 = np.array(X_test2)
    X_test2 = np.reshape(X_test2, (X_test2.shape[0], X_test2.shape[1], 1))
    pred_price2 = model.predict(X_test2)
    pred_price2 = scaler.inverse_transform(pred_price2)
    return " " + str(float(pred_price)) + " - " + str(float(pred_price2))

bebra = (make_prediction(selected_stocks, 'Close', False))
st.subheader('Використаємо Модель на основі LSTM для прогнозування

```

```

ціни')
st.write("Модель на основі LSTM має хороші результати на прогнозуванні
коротких проміжків часу")
st.write('Модель на основі LSTM дивиться на минулі 60 днів для того щоб
зробити прогноз на 61 день. ')
st.write("Вивидемо результат в таблиці та на графіку")
st.write(becca[0])
fig_tmp = go.Figure()
fig_tmp.add_trace(go.Scatter(x=becca[0]['Close'].index, y=becca[0]['Close'],
name='Реальна ціна на даних для тесту ',
line=dict(color='green')))
fig_tmp.add_trace(go.Scatter(x=becca[0]['Predictions'].index,
y=becca[0]['Predictions'], name='Прогнозована ціна',
line=dict(color='red')))
fig_tmp.add_trace(
go.Scatter(x=becca[1]['Close'].index, y=becca[1]['Close'], name='Реальна
ціна на даних для тренування',
line=dict(color='blue')))
fig_tmp.layout.update(xaxis_rangeslider_visible=True)
fig_tmp.update_layout(width=my_width, height=my_height)
st.plotly_chart(fig_tmp, width=my_width, height=my_height)

df_train = data[['Date', 'Close']]
df_train = df_train.rename(columns={"Date": "ds", "Close": "y"})

m = Prophet()
m.fit(df_train)

# прогноз на роки

st.subheader('Зробити прогноз від 1 до 5 років використовуючи Модель на
основі FbProphet')
st.write("Модель на основі FbProphet має хороші результати на
прогнозуванні довших проміжків часу")
st.write("Вивидемо результат в таблиці та на графіку")
n_years = st.slider("Кількість років для прогнозу:", 1, 5)
period = n_years * 365

```

```

future = m.make_future_dataframe(periods=period)
forecast = m.predict(future)

st.write(forecast)

fig1 = plot_plotly(m, forecast, xlabel="Time", ylabel="Price")
fig1.update_layout(width=my_width, height=my_height)
st.plotly_chart(fig1, width=my_width, height=my_height)

# прогноз на дні
st.subheader("Зробити прогноз від 1 до 100 днів використовуючи Модель
на основі FbProphet")
n_days = st.slider("Кількість днів для прогнозу:", 1, 100, value=20)
period = n_days

future_days = m.make_future_dataframe(periods=period)
forecast_days = m.predict(future_days)

st.write(forecast_days)

fig2 = plot_plotly(m, forecast_days, xlabel="Time", ylabel="Price")
fig2.update_layout(width=my_width, height=my_height)
st.plotly_chart(fig2, width=my_width, height=my_height)

# st.write('forecast data')
# fig2 = m.plot_components(forecast)
# st.write(fig2)

st.subheader("Зробити прогноз на конкретний день")
d = st.date_input(
    "Оберіть дату для прогнозу(тільки Пн-Пт)",
    datetime.date.today())
st.write('Результати:')
st.write('Model LSTM - Low-High = ', make_prediction(selected_stocks,
"Low", True, d))

forecast_days_new = m.predict(m.make_future_dataframe(periods=2000))
forecast_days_new['ds']
pd.to_datetime(forecast_days_new['ds']).apply(lambda x: x.date())

```

```

st.write('Model      Fbprophet      -      Low-High      =      ',
((forecast_days_new.loc[forecast_days_new['ds'] == d])['yhat_lower'].item()),
" - ",
      ((forecast_days_new.loc[forecast_days_new['ds'] ==
d])['yhat_upper'].item()))

```

```

def strategy_test(company_ticker, strategy):
    df = data_for_strategy
    training_data_len = math.ceil(len(df) * 0.7) # put 70% of our data to train
and 30 % to test with
    all_data = df[training_data_len:]
    # df = data_for_strategy
    # #training_data_len = math.ceil(len(df) * 0.7) # put 70% of our data to train
and 30 % to test with
    # all_data = df[:]
    res_close = make_prediction(company_ticker, 'Close', False)
    res_open = make_prediction(company_ticker, 'Open', False)

    delta_list = list()
    trade_dates = list()
    delta = 0.0
    for i in range(len(res_open[0])):

        if strategy == 1:
            delta = ((res_close[0]['Close'][i] - res_open[0]['Open'][i]) /
res_open[0]['Open'][i]) * 100
            delta_list.append(delta)
            trade_dates.append(res_close[0].index[i])

        if strategy == 2:
            if i > 1:
                if (((res_close[0]['Close'][i - 1] - all_data['Low'][i - 1]) /
all_data['Low'][i - 1]) * 100) < 0.75:
                    delta += ((res_close[0]['Close'][i] - res_open[0]['Open'][i]) /
res_open[0]['Open'][i]) * 100
                    delta_list.append(delta)
                    trade_dates.append(res_close[0].index[i])

```

```

if strategy == 3:
    if i >= 1:
        if (((res_open[0]['Open'][i - 1] - res_close[0]['Close'][i - 1]) /
            res_open[0]['Open'][i - 1]) * 100) > -1:
            delta += ((res_close[0]['Close'][i] - res_open[0]['Open'][i]) /
res_open[0]['Open'][i]) * 100
            delta_list.append(delta)
            trade_dates.append(res_close[0].index[i])

    if strategy == 4:
        if i >= 1:
            if (((all_data['Close'][i - 1] - all_data['Open'][i]) / all_data['Close'][i -
1]) * 100) > 3:
                if i + 20 > len(res_open[0]):
                    delta += ((res_close[0]['Close'][-1] - res_open[0]['Open'][i]) /
res_open[0]['Open'][i]) * 100
                    trade_dates.append(res_close[0].index[-1])
                else:
                    delta += ((res_close[0]['Close'][i + 19] - res_open[0]['Open'][i])
/
res_open[0]['Open'][i]) * 100
                    trade_dates.append(res_close[0].index[i + 19])
                    delta_list.append(delta)

fig_pnl = go.Figure()
fig_pnl.add_trace(
    go.Scatter(x=trade_dates, y=delta_list, name='Графік прибутку стратегії
',
                line=dict(color='green')))
fig_pnl.add_trace(go.Scatter(x=data["Date"], y=data["Close"],
name='Реальні дані',
                line=dict(color='blue')))

fig_pnl.layout.update(xaxis_rangeslider_visible=True)
fig_pnl.update_layout(width=my_width, height=my_height)
st.plotly_chart(fig_pnl, width=my_width, height=my_height)

```

```

return delta

st.subheader('Інформація по всім стратегіям :)')
st.write('Нижче наведено матрицю дохідності по всім стратегіям та опис кожної стратегії:')

image = Image.open('Figure_1.png')

st.image(image, width=1200, caption='Матриця дохідності в період з Березня 2020 по Травень 2022')

genre = st.radio(
    "Отримати опис стратегії",
    ('1', '2', '3', '4'))

st.write(" ")
if genre == '1':
    st.write('Стратегія полягає в тому, що якщо LSTM модель передбачає збільшення ціни акцій , то ми покупаємо і '
        'продаємо по прогнозу якщо до нього дійшло або по ціні закриття акції під кінець торгової '
        'сесії.Стратегія гарно працює на компаніях з великою капіталізацією.')
elif genre == '2':
    st.write("Це доповнення першої стратегії. Полягає в тому, що ми пропускаємо наш сигнал на покупку, якщо прогнозований зріст менше 1 %. Також гарно працює на компаніях з великою капіталізацією.")
elif genre == '3':
    st.write("Третя стратегія також виступає доповненням першої і одночасно другої. Полягає в тому, що якщо акції компанії впали на 1% за день і LSTM модель прогнозує збільшення більше ніж на 1%, то ми покупаємо і продаємо по прогнозу або по ціні закриття. ")
elif genre == '4':
    st.write("Четверта стратегія використовує модель FbProphet і націлена на компанії які мають тенденцію до росту. Суть стратегії полягає в тому щоб «відкупити» велике падіння якщо ми маємо прогноз на збільшення цін через місяць. Таким чином кожне падіння 5% та більше ми покупаем і продаем по прогнозу або ж через місяць, якщо до ціни прогнозу не доходить . ")

```



```
st.subheader("Обрати стратегію інвестування :")
st.write('про кожну стратегію ви можете прочитати вище')
strategys = (1, 2, 3, 4)
selected_strategy = st.selectbox("Оберіть стратегію : ", strategys)

data_load_state2 = st.text("Обробляємо дані для цієї стратегії ... ")
st.write('Результат стратегії : ', format(strategy_test(selected_stocks,
selected_strategy), '.2f'), '%')
data_load_state2.text("Дані Оброблено! ")
```