Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук
(повна назва)

Кафедра програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів штучного інтелекту для прогнозування динаміки акцій компанії за результатами аналізу корпоративних звітів
(тема)

Виконав:
студент 2 курсу, групи ІПЗМ-22-6

Границя А.В.
(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва спеціальності)
Тип програми освітньо-наукова

Керівник доц. кафедри ПІ Кравець Н.С.
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри _____ З.В.Дудар
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук
Кафедра програмної інженерії
Рівень вищої освіти другий (магістерський)
Спеціальність 121 – Інженерія програмного забезпечення
Тип програми освітньо-наукова програма
Освітня програма Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«___» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Границі Андрію Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження методів штучного інтелекту для прогнозування динаміки акцій компанії за результатами аналізу корпоративних звітів»
Затверджена наказом по університету від 29.03.2024р. № 250 Ст
2. Термін подання студентом роботи до екзаменаційної комісії 19.06.2024
3. Вихідні дані до роботи встановлений календарний план роботи, електронні ресурси за обраною тематикою, алгоритми прогнозування динаміки акцій, аналіз корпоративних звітів.
4. Перелік питань, що потрібно опрацювати в роботі *вступ, аналіз предметної галузі, постановка задачі, аналіз існуючих моделей, збір даних, реалізація моделей, проведення експерименту, аналіз результатів, висновки, перелік посилань, додатки*.
, висновки, перелік посилань, додатки.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі та постановка задачі	23.01 – 14.02.24	*виконано*
2	Аналіз та вибір API для дослідження	15.02 – 24.02.24	*виконано*
3	Аналіз та моделювання предметної області	17.02 – 28.02.24	*виконано*
4	Планування експериментів	25.02 – 28.02.24	*виконано*
5	Програмна реалізація кожної з обраних для дослідження моделей	25.02 – 01.04.24	*виконано*
6	Експериментальні дослідження	02.04 – 20.04.24	*виконано*
7	Аналіз результатів експериментальних досліджень та розробка рекомендацій	20.04 – 23.04.24	*виконано*
8	Написання та оформлення статті та тез доповіді	17.04 – 23.04.24	*виконано*
9	Підготовка пояснювальної записки	01.04 – 26.04.24	*виконано*
10	Підготовка презентації та доповіді	26.04 – 2.05.24	*виконано*
11	Нормоконтроль	09.06.24	*виконано*
12	Попередній захист	10.06.24	*виконано*
13	Занесення диплома в електронний архів	15.06.24	*виконано*
14	Рецензування	15.06.24	*виконано*
15	Допуск до захисту у зав. кафедри	18.06.24	*виконано*

Дата видачі завдання «*10*» *жовтня* 2023 р.

Студент _____ Границя А.В.

Керівник роботи _____ доцент каф. ПІ Кравець Н.С.

(підпис)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка містить: 71 стор., 16 рис., 2 табл., 19 джерел.

АКЦІЇ, АНАЛІЗ, ДОСЛІДЖЕННЯ, ГІБРИДНІ МОДЕЛІ, КОРПОРАТИВНІ ЗВІТИ, ПРОГНОЗУВАННЯ, ФІНАНСОВІ РИНКИ, ШТУЧНИЙ ІНТЕЛЕКТ, BERT, BIGBIRD, FINBERT, LSTM.

Об’єктом дослідження є методи штучного інтелекту для аналізу та прогнозування динаміки акцій компаній на основі корпоративних звітів.

Метою роботи є оцінка методів штучного інтелекту, які здатні ефективно аналізувати та прогнозувати напрямок зміни курсу акцій, використовуючи текстові звіти.

Методи дослідження базуються на застосуванні моделей штучного інтелекту, таких як BERT, BigBird, FinBERT та LSTM, для обробки великих обсягів фінансової інформації з корпоративних звітів.

В результаті роботи було проведено аналіз продуктивності моделей штучного інтелекту, визначено їх переваги та обмеження у фінансовому прогнозуванні, виявлено зв'язки між корпоративними звітами та змінами курсу акцій, а також розглянуто можливості покращення точності прогнозів на основі отриманих даних.

STOCKS, ANALYSIS, RESEARCH, HYBRID MODELS, CORPORATE REPORTS, FORECASTS, FINANCIAL MARKETS, ARTIFICIAL INTELLIGENCE, BERT, BIGBIRD, FINBERT, LSTM.

The object of the study is artificial intelligence methods for analyzing and forecasting the dynamics of company shares based on corporate reports.

The purpose of the work is to evaluate artificial intelligence methods that are capable of effectively analyzing and predicting the direction of stock price changes using text reports.

Research methods are based on the application of artificial intelligence models, such as BERT, BigBird, FinBERT and LSTM, to process large volumes of financial information from corporate reports.

As a result of the work, an analysis of the performance of artificial intelligence models was carried out, their advantages and limitations in financial forecasting were determined, connections between corporate reports and changes in stock prices were identified, and opportunities to improve the accuracy of forecasts based on the obtained data were also considered.

Я, Границя Андрій Володимирович, студент гр. ІПЗм-22-6, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження методів штучного інтелекту для прогнозування динаміки акцій компанії за результатами аналізу корпоративних звітів», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

ВСТУП

У сучасному світі швидкі технологічні зміни та постійний розвиток фінансових ринків вимагають постійного удосконалення методів аналізу та прогнозування динаміки акційних ринків. Застосування штучного інтелекту для аналізу фінансових показників та прогнозування руху цін на ринку стало одним з ключових напрямків у цій сфері. За останні роки відбувся значний прогрес у використанні методів штучного інтелекту, зокрема машинного та глибокого навчання, для аналізу фінансових даних та прогнозування ринкових тенденцій. Ці методи дозволяють автоматизувати процес аналізу великого обсягу інформації, виявляти складні зв'язки між різними факторами та робити прогнози з високою точністю.

Актуальність дослідження зумовлена не тільки швидким розвитком відповідних технологій, але й постійно зростаючою потребою фінансового сектора у новітніх інструментах аналізу та прогнозування.

Метою роботи є аналіз та дослідження методів штучного інтелекту для прогнозування динаміки акцій компаній, заснованих на аналізі корпоративних звітів. Для досягнення цієї мети досліджуються сучасні методи машинного навчання та обробки даних для створення прогнозних моделей, порівнюється ефективність різних підходів машинного навчання, включаючи рекурентні нейронні мережі, технології обробки природної мови (NLP) та гібридні моделі, з метою визначення найбільш ефективних стратегій для точного прогнозування рухів цін акцій на основі глибокого аналізу текстової інформації з корпоративних звітів.

Об’єктом дослідження є методи штучного інтелекту. Предметом дослідження є конкретні моделі та алгоритми штучного інтелекту, що застосовуються для прогнозування динаміки акцій компаній на основі текстової інформації з корпоративних звітів.

У дослідженні використовувалися такі методи: рекурентні нейронні мережі, технології обробки природної мови, гібридні моделі машинного навчання, а також статистичний аналіз для оцінки ефективності прогнозних моделей.

Наукові результати цієї роботи були представлені на XXVIII Міжнародному молодіжному форумі «Радіоелектроніка та молодь у XXI столітті» в рамках конференції «Інформаційні інтелектуальні системи». У доповіді «Оцінка ефективності використання методів штучного інтелекту для аналізу фінансових звітів та прогнозування динаміки акцій» було висвітлено результати дослідження моделей штучного інтелекту, зокрема рекурентних нейронних мереж та технологій обробки природної мови [1].

Результати цього дослідження можуть бути корисними для інвесторів, фахівців у галузі фінансів та управління портфелями, а також для компаній, які прагнуть зрозуміти та передбачити динаміку цін на їхні акції з максимальною точністю. Вони можуть бути впроваджені у практику фінансового аналізу для підвищення точності прогнозів та прийняття управлінських рішень.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Опис предментої області

Штучний інтелект (ШІ) – це область комп'ютерних наук, що займається створенням алгоритмів, які дозволяють машинам виконувати завдання, що традиційно вимагають людського інтелекту [2]. Особливо цінними є методи ШІ у сфері аналізу великих обсягів даних.

Предметна область дослідження включає в себе аналіз методів штучного інтелекту для прогнозування динаміки акцій компаній на основі аналізу корпоративних звітів. Ця тема є актуальною через значні обсяги даних, які генеруються компаніями та необхідність їх ефективного аналізу для прийняття обґрунтованих інвестиційних рішень.

В сучасних умовах фінансова аналітика все більше покладається на інноваційні технології для обробки та інтерпретації великих обсягів інформації. Використання алгоритмів штучного інтелекту дозволяє автоматизувати процес аналізу корпоративних звітів, знижуючи тим самим вплив людського чинника та підвищуючи точність прогнозів. Такі алгоритми можуть навчатися на великих наборах даних, виявляти приховані закономірності та робити передбачення на основі складних патернів у даних [3].

Корпоративні звіти містять різноманітну інформацію, включаючи фінансові показники, стратегії розвитку, аналіз ризиків, огляд ринку та інше. Аналіз цих звітів дає можливість зрозуміти поточний стан компанії та її перспективи, що є важливим для інвесторів та аналітиків. Проте, ручний аналіз великого обсягу таких даних є трудомістким і суб'єктивним, що робить використання ШІ особливо цінним.

Методи, які будуть досліджені в цій роботі, зосереджуються на використанні передових підходів штучного інтелекту, що дозволяють ефективно обробляти послідовні дані та виконувати завдання аналізу природної мови. Будуть досліджені різні стратегії та підходи, які комбінують найкращі практики в цій сфері для досягнення максимальної точності прогнозів.

Дослідження методів ШІ для прогнозування динаміки акцій має на меті не тільки підвищити точність прогнозів, але й зрозуміти, які саме аспекти

корпоративних звітів найбільше впливають на ринкову ціну акцій. Це дозволить інвесторам краще орієнтуватися у великому обсязі інформації та приймати більш обґрунтовані рішення. Також мета дослідження полягає в застосуванні комплексного аналізу, що поєднує різні методи обробки природної мови та прогнозування, оптимізовані для роботи з великими текстовими даними.

Таким чином, предметна область цього дослідження охоплює інтеграцію сучасних технологій ШІ, машинного та глибокого навчання, а також обробки природної мови для аналізу корпоративних звітів і прогнозування динаміки акцій. Цей підхід має потенціал суттєво вплинути на фінансову аналітику, зробивши процеси прогнозування більш точними, швидкими та об'єктивними.

1.2 Актуальність проблеми

Дослідження методів штучного інтелекту для прогнозування на основі текстових даних є важливим і актуальним напрямком у сфері фінансового аналізу. Цей напрямок залучає увагу через свою здатність використовувати складні алгоритми машинного навчання для ідентифікації закономірностей і тенденцій, які не завжди можуть бути виявлені традиційними методами аналізу.

На сьогоднішній день, з поширенням великих даних та вдосконаленням технологій, існує зростаюча потреба в розробці інноваційних підходів для аналізу фінансових ринків. Штучний інтелект, особливо з використанням методів обробки природної мови і нейронних мереж, дозволяє глибше аналізувати текстові корпоративні звіти, визначаючи ключові індикатори, які можуть впливати на ринкову вартість акцій.

Актуальність таких досліджень полягає у вирішенні проблеми ефективного аналізу великої кількості неструктурованих даних, які містяться у корпоративних звітах. Традиційні аналітичні методи часто не в змозі впоратися з таким об'ємом інформації або не можуть точно інтерпретувати складні семантичні і контекстуальні взаємозв'язки у текстах. В той же час, методи штучного інтелекту можуть автоматизувати цей процес, підвищуючи точність прогнозів і дозволяючи інвесторам та аналітикам приймати більш обґрунтовані рішення.

1.3 Аналіз існуючих досліджень

Аналіз існуючих рішень є важливою складовою дослідження. Це дозволяє оцінити сучасний стан технологій, виявити їхні переваги та недоліки, а також визначити напрямки для подальшого розвитку та інновацій.

Для аналізу було обрано низку наукових статей та досліджень, що стосуються використання методів штучного інтелекту для прогнозування фінансових показників компаній на основі текстових даних. Основна увага приділялась методам глибокого навчання, зокрема рекурентним нейронним мережам, та технологіям обробки природної мови через їхню здатність ефективно працювати з послідовними даними та складними текстовими структурами.

У дослідженні «S&P 500 Stock Price Prediction Using Technical, Fundamental and Text Data» було застосовано рекурентні нейронні мережі для аналізу послідовності даних у фінансових новинах [4]. Вони використовували LSTM (Long Short-Term Memory) для врахування часової послідовності даних, що є важливим для розуміння контексту у текстових даних. Використання LSTM дозволяє краще працювати зі складними текстовими даними, зменшуючи ризик втрати важливої інформації. Дослідження продемонструвало, що моделі на основі LSTM можуть ефективно враховувати історичні дані та їхній вплив на майбутні ціни акцій, що підтвердило високу точність прогнозів порівняно з традиційними методами машинного навчання.

Інше дослідження «News-Based Sparse Machine Learning Models for Adaptive Asset Pricing», зосередилося на використанні GRU (Gated Recurrent Units) для прогнозування динаміки акцій [5]. В їхньому підході поєднувалися як фінансові показники, так і текстові дані з новин, що дозволило враховувати додаткову інформацію про ринкові настрої та події, які можуть вплинути на ціни акцій. У цьому дослідженні модель GRU показала непогані результати точності.

У роботі «Using Financial News Sentiment for Stock Price Direction Prediction» було застосовано NLP модель FinBERT для аналізу фінансових новин [6]. FinBERT спеціально налаштована для фінансових текстів, дозволяє витягувати змістовні характеристики тексту, які потім використовуються для класифікації текстових

даних. Модель FinBERT показала високу ефективність у прогнозуванні змін цін акцій на основі аналізу даних з новин, показники точності моделі досягали 80 %. З цього дослідження випливає, що використання NLP моделей для прогнозування динаміки акцій на основі текстових даних має значний потенціал.

Переваги існуючих рішень:

– висока точність: Використання RNN та NLP дозволяє досягати високої точності прогнозів завдяки можливості аналізувати складні текстові дані та враховувати контекст;
– обробка великих обсягів даних: RNN та NLP моделі здатні обробляти великі масиви даних, що є критично важливим для аналізу корпоративних звітів;
– автоматизація процесів: Технології NLP та RNN дозволяють автоматизувати процес аналізу текстових документів, що значно зменшує витрати часу та ресурсів.

Недоліки:

– високі обчислювальні витрати: Навчання складних моделей RNN та NLP вимагає значних обчислювальних ресурсів;
– відсутність гібридних моделей: Поточні дослідження використовують окремо або RNN, або NLP та не поєднують їх у гібридні моделі, що може значно покращити точність і надійність прогнозів;
– відсутність передових моделей: Поточні дослідження не використовують новітні моделі, такі як BigBird та інші для обробки текстових даних саме для прогнозування динаміки акцій на основі корпоративних звітів, які можуть обробляти довші послідовності даних та забезпечувати більш глибокий контекстуальний аналіз.

Аналіз існуючих рішень показав, що сучасні методи штучного інтелекту, зокрема рекурентні нейронні мережі та технології обробки природної мови, мають значний потенціал для прогнозування на основі текстових даних.

Таким чином, дане дослідження має на меті подолання цих недоліків шляхом застосування та порівняння різних підходів машинного навчання, включаючи

гібридні моделі, що поєднують RNN та NLP, а також впровадження новітніх моделей для покращення точності прогнозів на основі глибокого аналізу текстової інформації з корпоративних звітів. Це дозволить визначити найбільш ефективні стратегії для точного прогнозування рухів цін акцій та сприятиме прийняттю більш обґрунтованих інвестиційних рішень.

1.4 Постановка задачі

Метою роботи є дослідження та аналіз методів штучного інтелекту для прогнозування динаміки акцій компаній на основі аналізу корпоративних звітів. У процесі аналізу предметної галузі, вивчення літератури та існуючих рішень були визначені основні напрями та завдання для подальшого дослідження. Ключовим завданням є пошук підходу, який забезпечить найвищу точність у прогнозуванні.

Для досягнення поставленої мети необхідно провести глибокий аналіз існуючих методів та підходів, що застосовуються у сфері штучного інтелекту для прогнозування фінансових ринків. Це включає вивчення переваг та недоліків сучасних методів обробки фінансових даних і текстової інформації, а також визначення найбільш перспективних алгоритмів машинного навчання та методів обробки природної мови.

Важливим етапом дослідження є збір та підготовка даних. Необхідно зібрати великий обсяг корпоративних звітів компаній та провести їх попередню обробку, що включає очищення, нормалізацію та перетворення текстових даних у формат придатний для навчання моделей. Визначення ключових показників та факторів, що впливають на динаміку акцій, також є критично важливим.

Після збору та підготовки даних, слід провести навчання та тестування існуючих моделей на історичних даних корпоративних звітів. Використання методів крос-валідації допоможе оцінити якість моделей та уникнути перенавчання. Тестування на нових, невідомих даних дозволить оцінити точність моделей і їх здатність прогнозувати динаміку акцій.

Проведений аналіз результатів прогнозування дозволить визначити найбільш ефективні підходи та методи для аналізу корпоративних звітів. Інтерпретація

отриманих результатів допоможе зрозуміти, які фактори найбільше впливають на точність прогнозів, і зробити висновки щодо ефективності наявних методів.

На завершальному етапі дослідження буде проведено всебічний аналіз результатів із метою визначення основних факторів, які впливають на точність прогнозів. Це дозволить ідентифікувати ключові аспекти, які потребують подальшого вдосконалення та розробки. Зокрема, буде розглянуто вплив різних архітектур нейронних мереж на точність прогнозів. Це допоможе визначити найбільш перспективні напрямки для майбутніх досліджень, включаючи інтеграцію додаткових джерел даних та розробку більш складних архітектур нейронних мереж.

Таким чином, задачі цього дослідження охоплюють комплексний аналіз існуючих методів штучного інтелекту для оцінки корпоративних звітів і прогнозування динаміки акцій.

2 АНАЛІЗ ІСНУЮЧИХ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ

2.1 Рекурентні нейронні мережі

Рекурентні нейронні мережі (RNN) - це клас штучних нейронних мереж, що можуть аналізувати послідовності даних, зберігаючи інформацію про попередні етапи обробки. Це створює внутрішній стан мережі, що дозволяє їй проявляти динамічну поведінку в часі [7]. RNN вважаються одними з найбільш ефективних інструментів для обробки послідовних даних, зокрема тексту. Це робить їх ідеальними для аналізу текстових звітів, де важливо враховувати контекстуальну інформацію. Розглянемо найпопулярніші RNN для завдань класифікації тексту.

LSTM (Long Short-Term Memory) - це спеціалізований тип рекурентних нейронних мереж, який ефективно вирішує проблему зникнення градієнту. Одна з найпопулярніших моделей у роботі з послідовними даними, особливо коли мова йде про довгострокові залежності в тексті. В архітектурі LSTM використовуються так звані "ворота", які контролюють потік інформації (див. рис. 2.1) Також нижче представлені усі основні формули LSTM [8].

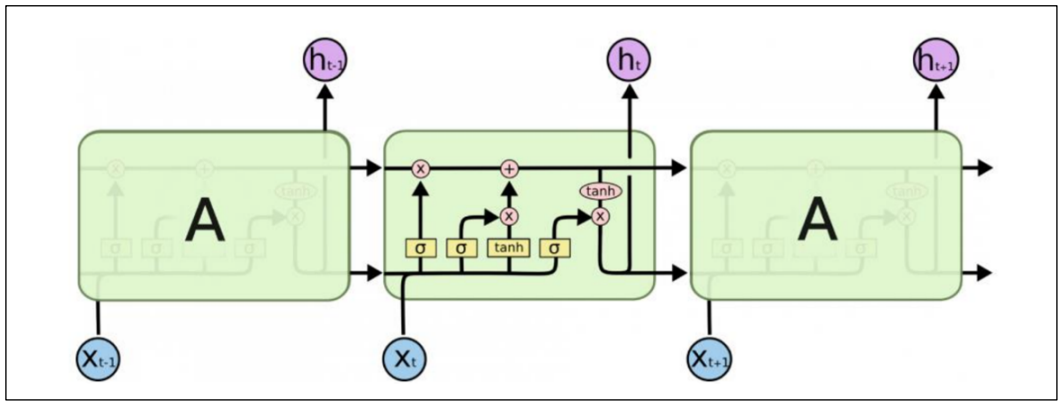

Рисунок 2.1 – Архітектура LSTM (за даними [9])

Ворота забування (Forget Gate): Визначає, яку інформацію слід викинути з комірки пам'яті. Формула 2.1:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2.1}$$

де σ – сигмоїдальна функція;

W_f – ваги воріт забування;

h_{t-1} – попередній прихований стан;

x_t – поточний вхідний вектор;

b_f – зміщення воріт забування.

Ворота входу (Input Gate): Визначає, яка нова інформація буде зберігатися в комірці пам'яті. Формула 2.2:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.2}$$

де σ – сигмоїдальна функція;

W_i – ваги воріт входу;

h_{t-1} – попередній прихований стан;

x_t – поточний вхідний вектор;

b_i – зміщення воріт входу.

Кандидатний стан блоку пам'яті c'_t можна описати наступною функцією (формула 2.3):

$$c'_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{2.3}$$

де *tanh* – гіперболічна тангенс функція;

W_c – ваги блоку пам'яті;

h_{t-1} – попередній прихований стан;

x_t – поточний вхідний вектор;

b_c – зміщення блоку пам'яті.

Стан блоку пам'яті c_t на поточному кроці можна описати функцією (формула 2.4):

$$c_t = f_t \cdot c_{t-1} + i_t \cdot c'_t \tag{2.4}$$

де f_t – вихід воріт забування;

c_{t-1} – попередній стан блоку пам'яті;

i_t – вихід воріт входу;

c'_t – оновлений кандидатний стан блоку пам'яті.

Ворота виходу (Output Gate): Визначає, яка частина стану комірки буде виведена у якості наступного прихованого стану. Формула:

Ворота виходу (Output Gate): Визначає, яка частина стану комірки буде виведена у якості наступного прихованого стану. Ця формула визначає ступінь, до якої поточний стан блоку пам'яті впливатиме на вихідний сигнал (формула 2.5):

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{2.5}$$

де σ – сигмоїдальна функція;

W_o – ваги воріт виходу;

h_{t-1} – попередній прихований стан;

x_t – поточний вхідний вектор;

b_o – зміщення воріт виходу.

Прихований стан h_t на поточному кроці можна описати наступною функцією. Ця формула визначає значення прихованого стану на основі виходу воріт виходу та поточного стану блоку пам'яті (формула 2.6):

$$h_t = o_t \cdot tanh(c_t) \tag{2.6}$$

де o_t – вихід воріт виходу;

tanh – гіперболічна тангенс функція;

c_t – поточний стан блоку пам'яті.

Ці ворота разом дозволяють LSTM моделювати як довгострокові, так і короткострокові залежності в даних. У прогнозування динаміки акцій, LSTM може ідентифікувати та використовувати важливі показники з фінансових звітів, які впливають на ціну акцій у довгостроковому періоді.

GRU (Gated Recurrent Unit) – подібна до LSTM, використовує принцип воріт для контролю потоку інформації, але з оптимізованою структурою, яка вимагає менше параметрів (див. рис. 2.2). Основними компонентами GRU є ворота оновлення (update gate) та ворота скидання (reset gate), також GRU не мають ворот виходу (output gate). Ворота визначають, як інформація з минулого стану впливає на поточний стан мережі. Усі основні формули для GRU наведено нижче [10].

Формула для воріт оновлення зводиться до виду (формула 2.7):

$$z(t) = \sigma(W_z \cdot x(t) + U_z \cdot h(t\text{-}1) + b_z) \tag{2.7}$$

де σ – сигмоїдальна функція;

U_z та W_z – вагові коефіцієнти воріт оновлення;

$h(t\text{-}1)$ – попередній стан;

$x(t)$ – поточний вхід;

b_z – зміщення воріт оновлення.

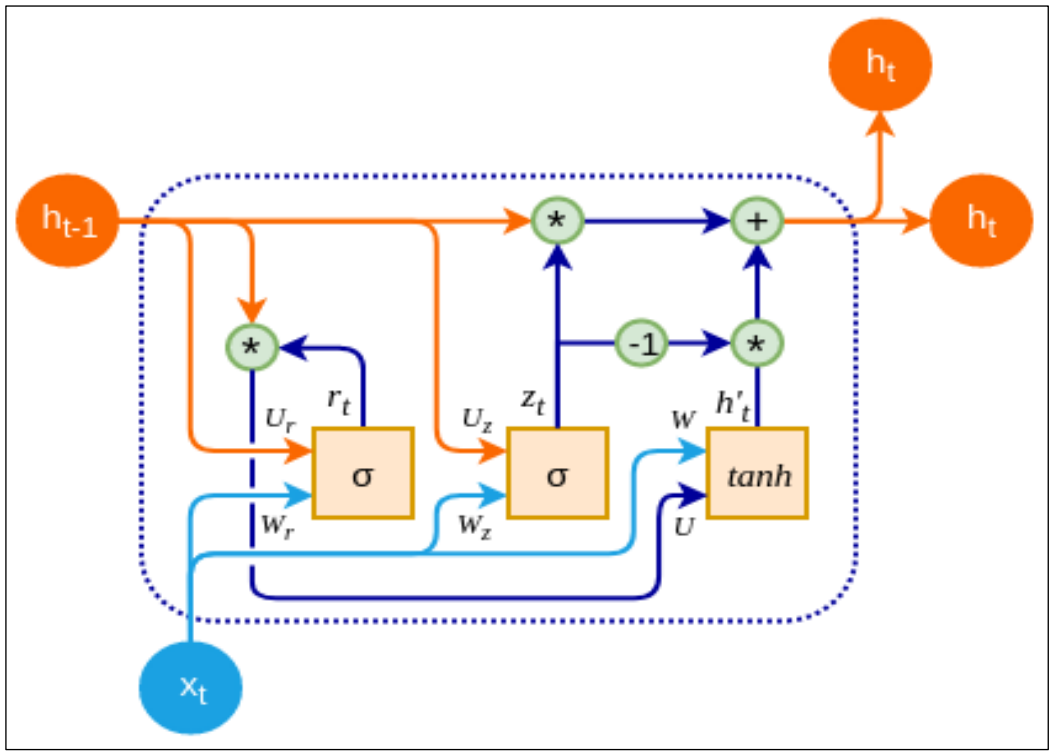

Рисунок 2.2 – Архітектура GRU (за даними [11])

Формула для воріт скидання виглядає наступним чином (формула 2.8):

$$r(t) = \sigma(W_r \cdot x(t) + U_r \cdot h(t\text{-}1) + b_r) \quad (2.8)$$

де σ – сигмоїдальна функція;

U_r та W_r – вагові коефіцієнти воріт скидання;

$h(t\text{-}1)$ – попередній стан;

$x(t)$ – поточний вхід;

b_r – зміщення воріт скидання.

Вихідне значення GRU обчислюється на основі проміжного стану $h'(\text{t})$, який враховує вплив попереднього стану згідно з воротами скидання (формула 2.9):

$$h'(\text{t}) = tanh(W_h \cdot x(t) + r(t) \cdot (U_r \cdot h(t\text{-}1)) + b_h) \quad (2.9)$$

де *tanh* – гіперболічна тангенс функція;

W_h – ваги поточного входу;

$x(t)$ – поточний вхідний вектор;

$r(t)$ – поточний вхідний вектор;

U_r – ваги попереднього прихованого стану;

$h(t\text{-}1)$ – попередній прихований стан;

b_h – зміщення.

Кінцевий стан $h(\text{t})$ визначається як комбінація проміжного стану і попереднього стану, використовуючи ворота оновлення (формула 2.10):

$$h(t) = z(t) \cdot h(t\text{-}1) + (1\text{-}\, z(t)) \cdot h'(\text{t}) \quad (2.10)$$

де $z(t)$ – вихід воріт оновлення;

$h(t\text{-}1)$ – попередній прихований стан;

$h'(\text{t})$ – проміжний стан блоку пам'яті;.

Таким чином, GRU ефективно вирішує проблеми зникнення або вибуху градієнту в рекурентних нейронних мережах, забезпечуючи збереження важливої інформації на тривалі періоди часу без значного збільшення обчислювальної

складності.

Обираючи між LSTM та GRU для аналізу текстових звітів, LSTM може виявитися більш відповідною з кількох причин. По-перше, здатність LSTM зберігати довготривалі залежності завдяки складній системі "воріт" є ключовою для розуміння складних і об'ємних текстів корпоративних звітів. Це дозволяє ефективніше обробляти довготривалі контексти. По-друге, хоча LSTM може вимагати більше часу та ресурсів для тренування через більшу кількість параметрів, її здатність знаходити та використовувати складні залежності у даних часто виправдовує ці витрати. Таким чином, вибір LSTM для задач дослідження є оптимальним, забезпечуючи баланс між глибиною аналізу та обчислювальною ефективністю, що є вирішальним для успішного прогнозування динаміки акцій.

2.2 NLP моделі

Обробка природної мови (NLP) є галуззю штучного інтелекту, яка зосереджена на взаємодії між комп'ютерами та людською (природною) мовою. Основна мета NLP полягає у забезпеченні можливості для комп'ютерів розуміти, інтерпретувати, генерувати та реагувати на людські мови в спосіб, який є цінним. Це включає різні задачі, такі як переклад, автоматична генерація текстів, визначення настроїв, класифікація текстів, розпізнавання мовлення та інші [12]. Розглянемо найпопулярніші NLP моделі для завдань класифікації тексту.

BERT (Bidirectional Encoder Representations from Transformers) - це одна з провідних моделей у сфері обробки природної мови, представлена у 2018 році. Модель була розроблена Google, і її створення стало значним кроком вперед у розвитку технологій обробки природної мови. Вона використовує механізм трансформерів, який дозволяє моделі ефективно аналізувати великі обсяги текстових даних (див. рис. 2.3). Особливість BERT полягає в тому, що вона аналізує контекст слова в обох напрямках (зліва направо та справа наліво), що робить її особливо потужною для розуміння контексту та семантики слів у тексті [13].

BERT була навчена на величезному корпусі тексту і може бути налаштована для конкретних завдань, таких як класифікація текстів, відповіді на питання, аналіз

настроїв тощо. У прогнозування динаміки акцій, BERT може аналізувати корпоративні звіти, новини про компанії та інші текстові дані, щоб виявляти закономірності та тренди, які можуть вплинути на ціну акцій.

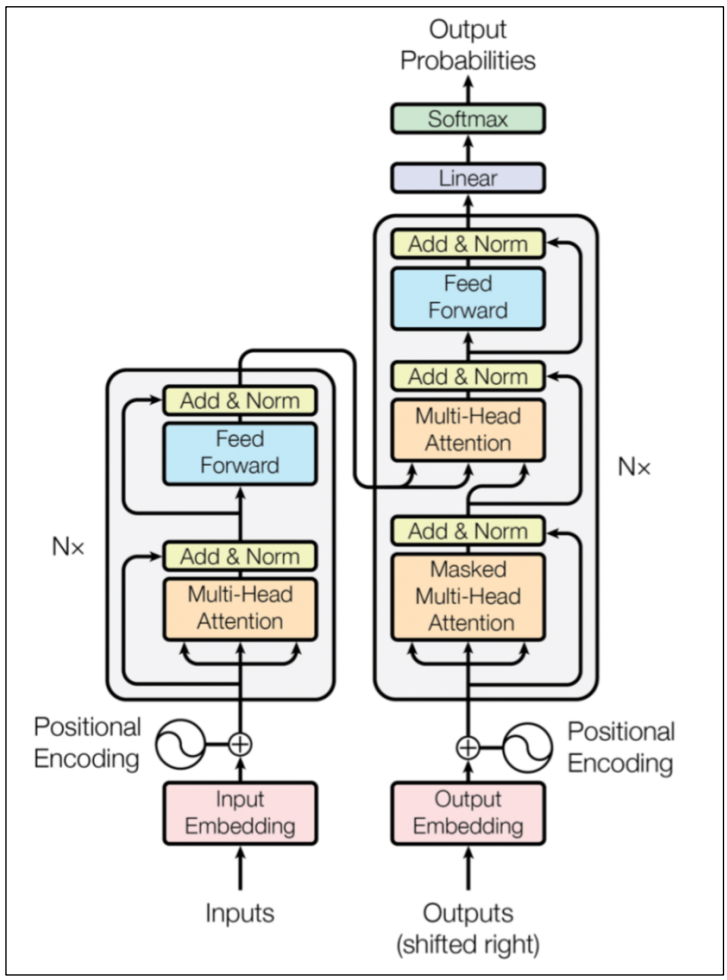

Рисунок 2.3 – Архітектура BERT (за даними [14])

Основний механізм BERT базується на трансформерах, які використовують увагу для моделювання залежностей між словами у тексті. Функція уваги визначається як (формула 2.11):

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (2.11)$$

де Q,K,V – це матриці запитів, ключів та значень;

d_k – розмірність ключів.

BERT використовує механізм багатоголової уваги, який дозволяє моделі

одночасно зосереджуватися на різних частинах тексту, що покращує її здатність розуміти складні контексти та семантичні відносини.

Після проходження через шари трансформера, вихід BERT може бути налаштований для виконання специфічних завдань. Наприклад, для класифікації текстів часто використовують додатковий лінійний шар, налаштований на основі вихідних даних BERT.

BERT, як передова модель обробки природної мови, є значним кроком вперед у розумінні та аналізі текстових даних. Завдяки своїй здатності розуміти контекст та семантику на глибокому рівні, BERT відкриває широкі можливості для різноманітних застосувань. Модель пропонує універсальність і точність, які роблять її ідеальним інструментом для сучасних завдань обробки мови.

BigBird розроблена як розширення моделі BERT, спроможна обробляти значно більші блоки тексту, ніж стандартний BERT. Це особливо важливо для аналізу довгих документів, які не можуть бути ефективно оброблені іншими моделями через обмеження на довжину вхідних даних. BigBird використовує модифіковану архітектуру трансформера, що дозволяє ефективно працювати з довгими послідовностями, забезпечуючи високу точність обробки [15].

FinBERT (Financial BERT) є спеціалізованою моделлю обробки природної мови, розробленою для аналізу фінансових текстів. Вона базується на архітектурі BERT і була навчена на великому корпусі фінансових даних, таких як новини, звіти та інші джерела фінансової інформації. FinBERT демонструє високу точність у задачах, пов'язаних з аналізом настроїв, класифікацією фінансових текстів і розпізнаванням сутностей. Завдяки спеціалізованому навчанню на фінансових даних, ця модель є особливо ефективною для завдань, які вимагають глибокого розуміння фінансового контексту [16].

Модель GPT (Generative Pre-trained Transformer) є потужною нейронною мережею, що базується на трансформерній архітектурі і тренується на величезних масивах даних. Вона здатна виконувати широкий спектр завдань обробки природної мови. Однак, у даному дослідженні GPT не використовується з кількох вагомих причин. По-перше, основна спеціалізація GPT полягає в генерації тексту, тоді як

завдання цього дослідження зосереджене на класифікації, яке є ключовим для прогнозування динаміки акцій. По-друге, передові версії GPT, такі як GPT-3 та GPT-4, не доступні для відкритого донавчання, що обмежує можливості їх застосування для спеціалізованих досліджень.

На основі аналізу представлених моделей для цілей дослідження були обрані BERT, BigBird та FinBERT. Вибір цих моделей зумовлений їхніми високими показниками у задачах аналізу та розуміння великих текстів. BERT ефективний у забезпеченні детального розуміння контексту, тоді як BigBird здатний обробляти особливо великі документи, а FinBERT, завдяки своєму навчанню на фінансових текстах, демонструє високу точність у завданнях, пов'язаних з аналізом фінансових даних.

2.3 Гібридні моделі

Гібридні моделі в машинному навчанні являють собою комбінації кількох алгоритмів або підходів, що використовуються разом для покращення точності, здатності до узагальнення та ефективності в різних задачах. Вони стають особливо важливими у складних задачах аналізу даних, де один алгоритм може не впоратися з різноманітністю та об'ємом інформації. У цьому розділі розглянемо популярні гібридні підходи класифікації тексту.

Перший підхід є комбінація методу вбудовування слів з рекурентними нейронними мережами (Word Embeddings + RNN). Word Embeddings використовуються для перетворення кожного слова або токена тексту в векторні представлення, що відображають його семантичні особливості. Ці векторні представлення слів подаються послідовно в рекурентну нейронну мережу, яка аналізує текст по кроку за кроком. В процесі обробки тексту, спочатку виконується вбудовування слів, після чого RNN аналізує кожен вектор, оновлюючи свій внутрішній стан на кожному кроці, що дозволяє моделі ухватити динаміку та залежності в тексті. Переваги цього підходу включають в себе здатність врахування послідовності інформації у тексті, а також здатність RNN вловлювати контекстуальні залежності між словами. Деякі з недоліків включають в себе

обмежену здатність RNN до вирішення проблеми з градієнтом на великих відстанях, а також обчислювальну складність роботи з цим типом моделей [17].

Другий підхід є комбінація трансформерів (найчастіше використовують BERT) з рекурентними нейронними мережами (BERT + RNN). BERT або інший трансформер можна використовувати для отримання контекстуалізованих векторних уявлень тексту. Тексти документів розбиваються на сегменти (якщо вони занадто довгі для одноразової обробки BERT), кожен сегмент обробляється моделлю для отримання векторних уявлень. BERT може обробляти тексти довжиною до 512 токенів, тому документи необхідно сегментувати відповідно. Векторні представлення від BERT подаються послідовно в RNN. RNN аналізує кожен вектор, оновлюючи свій внутрішній стан на кожному кроці, що дозволяє йому сприймати динаміку та залежності в даних протягом усього тексту. Після отримання ознак останній шар використовується для класифікації результату в одну з категорій. Переваги гібрида BERT + RNN може ефективно поєднувати глибоке розуміння контексту BERT з умінням вловлювати важливі локальні ознаки та часові залежності RNN [18].

Підхід Word Embeddings + RNN, в порівнянні з гібридом BERT + RNN, може бути менш ефективним у врахуванні глибокого розуміння контексту, оскільки вбудовання слів не можуть передавати так багато семантичної інформації, як BERT. Тому, для проведення дослідження буде використана гібридна модель, що поєднує трансформер із комбінацією рекурентних нейронних мереж, а саме - BERT + LSTM. Обґрунтування використання цієї моделі полягає у тому, що BERT забезпечує глибоке розуміння контексту тексту, тоді як LSTM може ефективно аналізувати послідовності даних і враховувати довготривалі залежності.

3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ

3.1 Збір даних

Вибір надійного джерела даних і їх ефективна обробка є необхідними для точного та ефективного навчання нейронних мереж. Збір даних передбачає отримання, структурування та адаптацію інформації для подальшого використання у машинному навчанні. Важливо забезпечити, щоб зібрані дані були репрезентативними, достатньо об'ємними та точними, оскільки це безпосередньо впливає на якість навчання та результативність моделі.

Після детального аналізу існуючих корпоративних звітів компаній, було вирішено зосередити дослідження на звітах, які мають форму 10-Q. Форма 10-Q – це квартальний звіт американських публічних компаній, поданий до Комісії з цінних паперів та бірж, що містить детальну інформацію про фінансовий стан, результати діяльності та ризики компанії. Ця форма була обрана через її високий рівень деталізації та структурованість інформації, і тому ідеально підходить для навчання моделей.

Для дослідження були обрані саме американські компанії, оскільки Сполучені Штати мають один з найбільш розвинених фінансових ринків у світі з високою ступенем прозорості корпоративного звітування та великою кількістю компаній, які регулярно подають звіти. Це забезпечує великий обсяг даних із порівняно стабільним та передбачуваним економічним та законодавчим середовищем, що є критично важливим для точності дослідження. Наявність цих даних дозволяє проводити глибокий статистичний аналіз і тренування прогностичних моделей штучного інтелекту з вищою точністю та надійністю прогнозів.

При виборі джерел даних для дослідження, особлива увага була приділена якості та доступності інформації. Одним із ключових ресурсів, для збору даних, було обрано SEC-API.io [19]. Цей ресурс вибрано завдяки його високій структурованості даних та зручності використання. SEC-API.io надає доступ до обширної бази даних корпоративних звітів компаній, зокрема до квартальних звітів форми 10-Q. Також сервіс надає текст звіту з розбиттям на секції, що забезпечує додаткові можливості для тренування моделей.

Для цього дослідження було обрано 15 найвідоміших американських компаній, з яких було зібрано звіти за період з 1994 по 2024 рік. Всього було зібрано 1300 звітів, які в свою чергу містять близько 9600 секцій.

Також для повноцінного прогнозування динаміки акцій компаній необхідно мати дані курсу акцій компаній за день до публікації звітів, в день публікації та в день після публікації. З цією метою було використано дані з сервісу Yahoo Finance, який є відомим провайдером фінансової інформації. Цей сервіс надає широкий спектр даних про фінансові ринки, включно з історичними та поточними цінами акцій. Для збору даних було використано офіційну бібліотеку yfinance написану на Python.

В результаті обробки зібраних даних було створено дві взаємопов'язані таблиці: reports та sections. Таблиця reports є основною і містить наступні поля:

– id – унікальний індентифікатор звіту;
– ticker – відображає скорочену назву компанії;
– company_name – відображає повну назву компанії;
– filled_at – дата заповнення та публікації звіту;
– form_type – тип звіту;
– report_url – посилання на звіт;
– close_price_day_before – курс акцій за день до публікації звіту;
– close_price_day_of_report – курс акцій у день публікації;
– close_price_day_after – курс акцій наступного дня після публікації.

Таблиця sections служить для зберігання даних про окремі секції звітів і включає поля:

– id – унікальний індентифікатор секції;
– text – текст секції звіту;
– section – назва розділу звіту;
– report_id – індентифікатор звіту, до якого належить секція.

Взаємозв'язок між таблицями reports та sections реалізований за принципом "один до багатьох", де один запис у таблиці reports може бути асоційований з багатьма записами у таблиці sections, що дозволяє ефективно організувати та

аналізувати інформацію з корпоративних звітів.

3.2 Попередня обробка даних

У рамках дослідження, важливим етапом є попередня обробка даних. Першочергово було проведено очищення текстів звітів від невідомих символів, зайвих відступів та переносів рядків. Для цього була розроблена спеціалізована функція обробки текстів. Цей процес включав використання регулярних виразів та текстових фільтрів для видалення зайвих даних.

Центральним аспектом попередньої обробки було формування цільової змінної stock_trend, що відображає тренди зміни цін акцій. Цільова змінна визначалась на основі процентної різниці між курсами акцій за день до та після публікації звіту. Використовуючи дані про ціни акцій, отримані з Yahoo Finance, розрахунок полягав у визначенні відсоткової зміни ціни, на основі якої stock_trend класифікувався як:

- Positive, якщо змінення курсу становило більше ніж 5%;
- Negative, якщо змінення курсу становило менше ніж -5%;
- Neutral, у всіх інших випадках.

Поріг у 5% для зміни курсу акцій був обраний як достатній для класифікації звітів з декількох причин. По-перше, зміни ціни акцій на рівні 5% і більше часто вважаються суттєвими на фінансових ринках, оскільки вони можуть відображати важливі події або новини, що впливають на курс. По-друге, такий поріг допомагає виділити більш виразні сигнали серед щоденних коливань ринку, що підвищує точність прогнозів та знижує вплив волатильності на ринку. Нарешті, використання 5% порогу узгоджується з практикою інших досліджень у галузі фінансового аналізу, що дозволяє порівнювати результати даного дослідження з іншими роботами у цій сфері.

Після обробки текстів і розрахунку цільової змінної було сформовано датасет, який містить всі необхідні дані для тренування моделей. Основною змінною для навчання моделей є перемінна processed_text, що включає текст корпоративного звіту у вигляді масиву секцій. Цільова змінна, яка розраховується з перемінної

stock_trend, має три значення: 0 – Neutral, 1 – Positive, 2 – Negative. Датасет також містить основну інформацію зі звітів (див. рис. 3.1). Для забезпечення узгодженості та надійності результатів, дані були ретельно перевірені на наявність пропусків та аномалій. Розмір сформованого датасету складає 946 елементів. Цей датасет є основою для подальшого аналізу та тренування моделей.

	processed_text	ticker	company_name	form_type	filed_at	percent_change	stock_trend
0	['Item 1 Financial Statements ...	XOM	EXXON MOBIL CORP	10-Q	2016-08-03 16:10:47	0.505518	Neutral
1	['Item 1 Financial Statements ...	XOM	EXXON MOBIL CORP	10-Q	2014-05-07 16:02:50	-0.340897	Neutral
2	['Item 1 Financial Statements ...	XOM	EXXON MOBIL CORP	10-Q	2017-05-03 18:32:17	-0.499700	Neutral
3	['Item 1 Financial Statements EXXON MOBIL CORP...	XOM	EXXON MOBIL CORP	10-Q	2010-05-06 17:53:44	-3.732806	Neutral
4	['PART I FINANCIAL INFORMATION ITEM 1 FINANCIA...	XOM	EXXON MOBIL CORP	10-Q	2023-05-02 17:24:25	-5.877734	Negative

Рисунок 3.1 – Приклад даних датасету (рисунок виконаний самостійно)

Таким чином, етап обробки даних забезпечує належну підготовку даних, необхідну для ефективного застосування алгоритмів машинного навчання, що відкриває широкі можливості для подальших досліджень та практичного застосування у фінансовому секторі.

3.3 План-програма експерименту

Для реалізації експериментальної частини дослідження було обрано пристрій на базі MacOS. Технічні характеристики даного пристрою наведені у таблиці 3.1.

Для реалізації моделей була обрана мова програмування Python версії 3.12 та середовище розробки PyCharm версії 17.0.3. Також були використані наступні бібліотеки:

- tensorflow 2.11.0;
- transformers 4.39.1;
- torch 2.2.1;
- yfinance 0.2.37;
- sec_api 1.0.18;
- numpy 1.26.4;

– matplotlib 3.8.4;
– seaborn 0.13.2;
– pandas 2.2.1;
– scikit-learn 1.4.1.

Таблиця 3.1 – Характеристики пристрою (таблиця виконана самостійно)

Характеристика	Значення
Name	MacBook Pro 16 (2021)
CPU	Apple M1
RAM	16 GB
OS	macOS
SSD	512 GB

В рамках експерименту потрібно зібрати числові дані для всіх показників кожного використаного методу. Для аналізу ефективності та порівняння алгоритмів LSTM, BERT, FinBERT, BigBird та BERT + LSTM будуть визначені такі ключові метрики:

– accuracy є одним із найпоширеніших показників ефективності моделей машинного навчання, особливо в задачах класифікації. Вона визначає частку правильних передбачень серед усіх зроблених передбачень. Точність є корисним показником у випадках, коли кількість позитивних і негативних класів приблизно однакова;
– precision – точність передбачень, або позитивна передбачувальна цінність, вимірює частку правильно передбачених позитивних випадків серед усіх передбачених позитивних випадків. Цей показник є важливим у ситуаціях, де критичним є уникнення помилкових спрацьовувань. Висока точність

передбачень свідчить про те, що модель рідко помилково ідентифікує негативні випадки як позитивні;

– recall – повнота, також відома як чутливість, визначає здатність моделі виявляти всі релевантні позитивні випадки. Вона вимірює частку правильно ідентифікованих позитивних випадків серед усіх фактичних позитивних випадків. Висока повнота вказує на те, що модель рідко пропускає позитивні випадки;
– f1-Score є комбінованим показником, який поєднує точність передбачень та повноту, забезпечуючи збалансовану оцінку ефективності моделі. Він особливо корисний в ситуаціях, коли важливо враховувати як точність передбачень, так і повноту, і коли існує значний дисбаланс між класами. Високий F1-Score свідчить про те, що модель має хороший баланс між точністю та повнотою передбачень;
– час прогнозування – це метрика, яка вимірює, скільки часу потрібно моделі для здійснення передбачення. Цей показник важливий для оцінки продуктивності та придатності моделей для застосувань у реальному часі;
– час навчання моделі – це показник, який вимірює, скільки часу потрібно для навчання моделі. Ця метрика є важливою для оцінки загальної ефективності моделі та ресурсів, необхідних для її навчання.

4 РЕАЛІЗАЦІЯ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ

4.1 Реалізація LSTM

Розглянемо основні кроки реалізації моделі LSTM. Важливим етапом в цьому процесі є підготовка даних, що була виконана у попередньому розділі дослідження.

Токенізація текстів здійснювалася із застосуванням бібліотеки Keras, яка перетворює текст на послідовності індексів слова з обмеженим словарем 10 000 найбільш частотних слів. Це дозволяє зменшити розмір векторного простору, що значно спрощує обчислювальні вимоги для моделі. Після токенізації кожен текст був приведений до єдиної довжини з використанням методу pad_sequences, що забезпечує однакові умови для обробки всіх текстів у нейронній мережі. Код токенізації LSTM моделі на мові програмування Python:

```
tokenizer = Tokenizer(num_words=10000)
tokenizer.fit_on_texts(result['processed_text'])
sequences = tokenizer.texts_to_sequences(result['processed_text'])
X = pad_sequences(sequences, maxlen=2000)
```

Модель LSTM була реалізована з використанням послідовного підходу бібліотеки Keras. Нижче наведено код створення LSTM моделі на мові програмування Python:

```
model = Sequential()
model.add(Embedding(10000, 128, input_length=2000))
model.add(Bidirectional(LSTM(64, dropout=0.2, recurrent_dropout=0.2,
kernel_regularizer=l2(0.01))))
model.add(Dropout(0.2))
model.add(Dense(64, activation='relu', kernel_regularizer=l2(0.01)))
model.add(Dropout(0.2))
model.add(Dense(3, activation='softmax', kernel_regularizer=l2(0.01)))
model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])
```

Модель включає кілька ключових компонентів:

– embedding шар: перетворює індекси слів у вектори фіксованого розміру. Цей шар дозволяє моделі навчатися відносинам між словами, забезпечуючи кращу семантичну обробку тексту;

- bidirectional LSTM шар: LSTM шар дозволяє моделі зберігати важливу інформацію з довгих послідовностей тексту, враховуючи контекст з минулого. Використання двонаправленого LSTM забезпечує додаткову точність, оскільки модель аналізує текст у двох напрямках. Це дозволяє моделі краще розуміти залежності між словами в тексті, незалежно від того, де ці слова розташовані у реченні;
- dropout шари: використовуються для запобігання перенавчанню моделі. Ці шари випадковим чином відключають певний відсоток нейронів під час кожної ітерації тренування, що сприяє зниженню залежності моделі від конкретних нейронів і підвищує її здатність до узагальнення. Таким чином, модель стає більш стійкою до перенавчання на навчальних даних, що покращує її продуктивність на нових, невідомих даних;
- dense шари: це шари, які використовують функції активації ReLU та softmax. Функція активації ReLU (Rectified Linear Unit) додає нелінійність моделі, що дозволяє їй ефективніше вловлювати складні залежності в даних. Функція активації softmax, яка застосовується на вихідному шарі, дозволяє моделі виконувати багатокласову класифікацію.

Навчання моделі проводилося з використанням функції ранньої зупинки для запобігання перенавчанню. Функція ранньої зупинки припиняє тренування, якщо модель не покращується протягом трьох епох:

```
early_stopping = EarlyStopping(monitor='val_loss', patience=3,
restore_best_weights=True)
history = model.fit(X_train, y_train, batch_size=32, epochs=20,
validation_split=0.1, verbose=1, callbacks=[early_stopping])
```

Цей підхід дозволяє моделі зберегти найкращі ваги, що знижує ризик перенавчання та підвищує загальну ефективність.

Після тренування, модель оцінюється на тестових даних для визначення її точності. Це дозволяє оцінити здатність моделі до коректного прогнозування.

4.2 Реалізація BERT та FinBERT

У цьому розділі представлено покрокову реалізацію використання моделі BERT та FinBERT для класифікації текстів корпоративних звітів. Для роботи з моделями використовується бібліотека transformers, яка надає засоби для легкої інтеграції моделі у Python.

На першому етапі необхідно завантажити дані з датасету, які містять тексти корпоративних звітів та відповідні мітки класів. Після цього дані розділяються на навчальну та тестову вибірки.

Наступним кроком є ініціалізація токенізатора та моделі. Токенізатор відповідає за перетворення тексту у формат, придатний для обробки моделлю. Наведемо програмну реалізацію для моделі BERT.

```
from transformers import BertTokenizer,   BertForSequenceClassification
import pandas as pd
# Ініціалізація токенізатора та моделі BERT
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
model = BertForSequenceClassification.from_pretrained('bert-base-
uncased', num_labels=3)
```

У цьому прикладі було використано токенізатор bert-base-uncased, який перетворює текст у послідовність токенів. Модель BertForSequenceClassification була попередньо навчена і налаштована для класифікації текстів на три класи (вказані параметром num_labels=3).

Для ініціалізації токенізатора та моделі FinBERT виконується аналогічний процес, тільки замість bert-base-uncased було використано yiyanghkust/finbert-tone.

```
from transformers import BertTokenizer,   BertForSequenceClassification
# Ініціалізація токенізатора та моделі FinBERT
tokenizer_finbert = BertTokenizer.from_pretrained('yiyanghkust/finbert-
tone')
model_finbert =
BertForSequenceClassification.from_pretrained('yiyanghkust/finbert-
tone', num_labels=3)
```

Одним із ключових викликів при використанні цих моделей є обмеження на довжину вхідного тексту до 512 символів. Це створює проблему, коли середня

довжина корпоративного звіту становить близько 60000 символів. Для вирішення цієї проблеми звіти розбиваються на секції, кожна з яких обробляється окремо, а потім комбінуються для токенізації. Для підготовки даних використовується клас ReportDataset, який відповідає за токенізацію та перетворення тексту у формат, придатний для моделі:

```
class ReportDataset(Dataset):
    def __init__(self, reports, targets, tokenizer, max_len):
        self.reports = reports
        self.targets = targets
        self.tokenizer = tokenizer
        self.max_len = max_len
    def __getitem__(self, item):
        report = eval(self.reports[item])  # Перетворюємо рядок на
список секцій
        target = self.targets[item]
        # Об'єднуємо всі секції в один довгий текст
        combined_report = " ".join(report)
        # Токенізація об'єднаного тексту
        encoding = self.tokenizer.encode_plus(
            combined_report,
            add_special_tokens=True,
            max_length=self.max_len,
            truncation=True,
            padding='max_length',
            return_attention_mask=True,
            return_tensors='pt',
        )
        return {
            'input_ids': encoding['input_ids'].flatten(),
            'attention_mask': encoding['attention_mask'].flatten(),
            'labels': torch.tensor(target, dtype=torch.long)
        }
```

У цьому коді кожен звіт розбивається на кілька секцій, які потім об'єднуються в один довгий текст. Цей текст токенізується з урахуванням обмеження в 512 символів. Це дозволяє обробляти великі документи частинами, зберігаючи при цьому контекст і релевантну інформацію.

Основні параметри тренування моделі задаються у TrainingArguments, після чого створюється об'єкт Trainer, який відповідає за тренування моделі:

```
# Налаштування для тренування моделі
training_args = TrainingArguments(
    output_dir='./results',
```

```
    save_total_limit=2,
    save_steps=500,
    evaluation_strategy="steps",
    eval_steps=500,
    logging_dir='./logs',
    logging_steps=10,
    per_device_train_batch_size=BATCH_SIZE,
    per_device_eval_batch_size=BATCH_SIZE,
    learning_rate=2e-5,
    warmup_steps=100,
    weight_decay=0.01,
    load_best_model_at_end=True,
    metric_for_best_model="eval_loss",
    greater_is_better=False,
    num_train_epochs=10
)
data_collator = DataCollatorWithPadding(tokenizer)
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_data_loader.dataset,
    eval_dataset=val_data_loader.dataset,
    compute_metrics=compute_metrics,
    data_collator=data_collator,
  callbacks=[EarlyStoppingCallback(early_stopping_patience=2)]
)
# Навчання моделі
trainer.train()
```

Параметри тренування включають частоту збереження моделі, оцінку ефективності, швидкість навчання, кількість епох та використання ранньої зупинки для уникнення перенавчання.

Після тренування моделей проводиться аналіз їх ефективності. Важливо оцінити точність моделей у класифікації текстів та її здатність виявляти ключові тренди та інформацію, яка може вплинути на динаміку акцій. Це включає аналіз помилок класифікації та оцінку, як модель справляється з різними типами корпоративних звітів.

4.3 Реалізація BigBird

У цьому розділі представлено покрокову реалізацію використання моделі BigBird для класифікації текстових звітів. Спочатку необхідно завантажити дані з датасету, що містять тексти корпоративних звітів та відповідні мітки класів. Потім дані розподіляються на навчальну та тестову вибірки.

Після завантаження даних слід ініціалізувати токенізатор та модель BigBird.

Ця модель оптимізована для роботи з довгими текстами, що є перевагою при обробці корпоративних звітів. Використовується токенізатор BigBirdTokenizer та модель BigBirdForSequenceClassification, які були попередньо навчені та налаштовані для завдань класифікації тексту. Важливо також включити функцію gradient checkpointing для ефективного використання пам'яті під час тренування:

```
tokenizer = BigBirdTokenizer.from_pretrained('google/bigbird-roberta-
base')
model = BigBirdForSequenceClassification.from_pretrained('
google/bigbird-roberta-base', num_labels=3)
model.gradient_checkpointing_enable()
```

Для обробки текстів корпоративних звітів використовується клас ReportDataset. Тексти розбиваються на сегменти, кожен з яких обробляється окремо, щоб обмеження на кількість токенів у моделі BigBird не перевищувалося. Клас здійснює токенізацію кожного сегменту тексту та підготовку даних для моделі.

Наступним кроком є налаштування параметрів тренування моделі. Використовується TrainingArguments для визначення основних параметрів, таких як частота збереження моделі, частота оцінки, швидкість навчання, кількість епох тощо. Для тренування моделі використовується Trainer, який відповідає за організацію процесу тренування, збереження кращих моделей і оцінку ефективності.

Після завершення тренування моделі проводиться оцінка її ефективності на тестових даних. Отримані результати дозволяють оцінити здатність моделі правильно класифікувати тексти корпоративних звітів.

4.4 Реалізація BERT + LSTM

Розглянемо детальну реалізацію гібридної моделі BERT + LSTM. Перед тренуванням моделі, датасет з корпоративних звітів був підготовлений шляхом токенізації текстів із застосуванням токенізатора від BERT. Він перетворює текст на послідовність токенів, додаючи спеціальні символи для позначення початку та кінця послідовності. Це дозволяє ефективно обробляти тексти незалежно від їхньої довжини, розбиваючи довгі тексти на сегменти довжиною до 512 токенів.

Гібридна модель складається з двох основних компонентів: BERT та LSTM. BERT використовується для отримання контекстуалізованих векторних уявлень тексту, після чого ці векторні уявлення передаються до LSTM. LSTM обробляє ці уявлення, враховуючи послідовність та довготривалі залежності між словами, що дозволяє отримати більш точні результати класифікації. Нижче наведено фрагмент коду реалізації гібридної моделі.

```
class BERT_LSTM(nn.Module):
    def __init__(self, bert, hidden_dim, num_labels):
        super(BERT_LSTM, self).__init__()
        self.bert = bert
        self.lstm = nn.LSTM(
            input_size=bert.config.hidden_size,
            hidden_size=hidden_dim,
            batch_first=True,
            bidirectional=True
        )
        self.classifier = nn.Linear(hidden_dim * 2, num_labels)  #
Удвоєння hidden_dim через bidirectional LSTM
    def forward(self, input_ids, attention_mask, labels=None):
        # Отримання контекстуалізованих уявлень від BERT
        with torch.no_grad():
            outputs = self.bert(input_ids=input_ids,
attention_mask=attention_mask)
        last_hidden_state = outputs[0]
        # Обробка уявлень через LSTM
        lstm_out, _ = self.lstm(last_hidden_state)
        logits = self.classifier(lstm_out[:, -1, :])
        # Обчислення функції втрат
        loss = None
        if labels is not None:
            loss_fct = nn.CrossEntropyLoss()
            loss = loss_fct(logits.view(-1,
self.classifier.out_features), labels.view(-1))
        return (loss, logits)
```

Клас BERT_LSTM інтегрує можливості моделей BERT та LSTM, забезпечуючи ефективну обробку та класифікацію тексту. У конструкторі класу ініціалізуються модель BERT, LSTM та класифікаційний шар. Модель BERT відповідає за створення контекстуалізованих уявлень тексту, LSTM обробляє ці уявлення з урахуванням послідовності слів, а класифікаційний шар виконує остаточну класифікацію.

Метод forward приймає на вхід ідентифікатори токенів та маски уваги, а

також, мітки. Спочатку вхідні дані передаються через модель BERT, яка у свою чергу генерує контекстуалізовані векторні уявлення кожного токена. Цей процес відбувається без обчислення градієнтів, що дозволяє зекономити пам'ять і ресурси.

Отримані від BERT векторні уявлення передаються до LSTM. Рекурентна нейронна мережа LSTM обробляє ці уявлення, враховуючи послідовні залежності між словами. Потім вихід LSTM передається через лінійний класифікаційний шар, який генерує логіти для кожної категорії. Також обчислюється функція втрат за допомогою CrossEntropyLoss, яка використовується для навчання моделі. Метод forward повертає логіти та функцію втрат.

Після ініціалізації моделі визначаються параметри тренування за допомогою TrainingArguments, які включають шлях до директорії для збереження результатів, частоту збереження моделі, стратегію оцінювання, частоту оцінки, розмір батчу, швидкість навчання, кількість кроків для розігріву, вагове затухання, завантаження найкращої моделі в кінці та кількість епох тренування. Також використовується DataCollatorWithPadding для додаткової обробки вхідних даних.

Тренування моделі здійснюється за допомогою об'єкта Trainer, який відповідає за тренування моделі на навчальному наборі даних, оцінку на валідаційному наборі та обчислення метрик. Після завершення тренування модель оцінюється на тестовій вибірці для отримання основних метрик ефективності.

5 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТУ

5.1 Модель LSTM

Модель LSTM пройшла навчання за 10 епох. Час, необхідний для навчання моделі, становить 406.42 секунди, а для прогнозування – 1.09 секунди. Швидкість навчання є досить високою, що дозволяє швидко адаптувати модель до нових даних, а швидкість прогнозування забезпечує майже миттєву обробку результатів. Графік функції втрат моделі LSTM демонструє постійне зниження втрат з кожною епохою, що свідчить про поступове наближення моделі до оптимального стану (див. рис. 5.1).

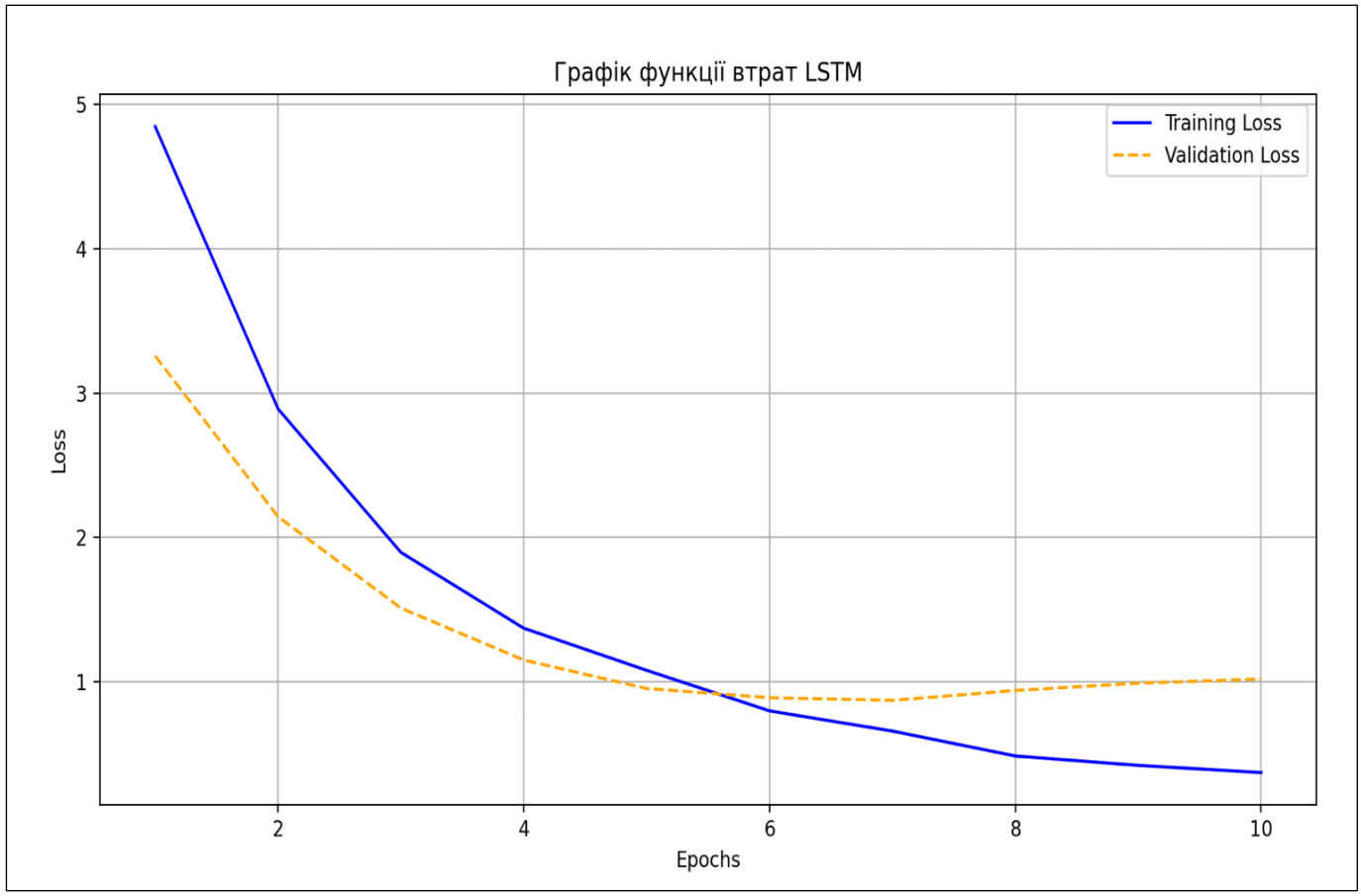

Рисунок 5.1 – Графік loss функції моделі LSTM (рисунок виконаний самостійно)

Для оцінки якості роботи моделі використовувалися метрики accuracy, precision, recall та F1-Score. Графік нижче ілюструє зміну цих метрик протягом епох (див. рис. 5.2). Значення метрик збільшуються з кожною епохою, що свідчить про здатність моделі адаптуватися до вхідних даних та покращувати свої показники.

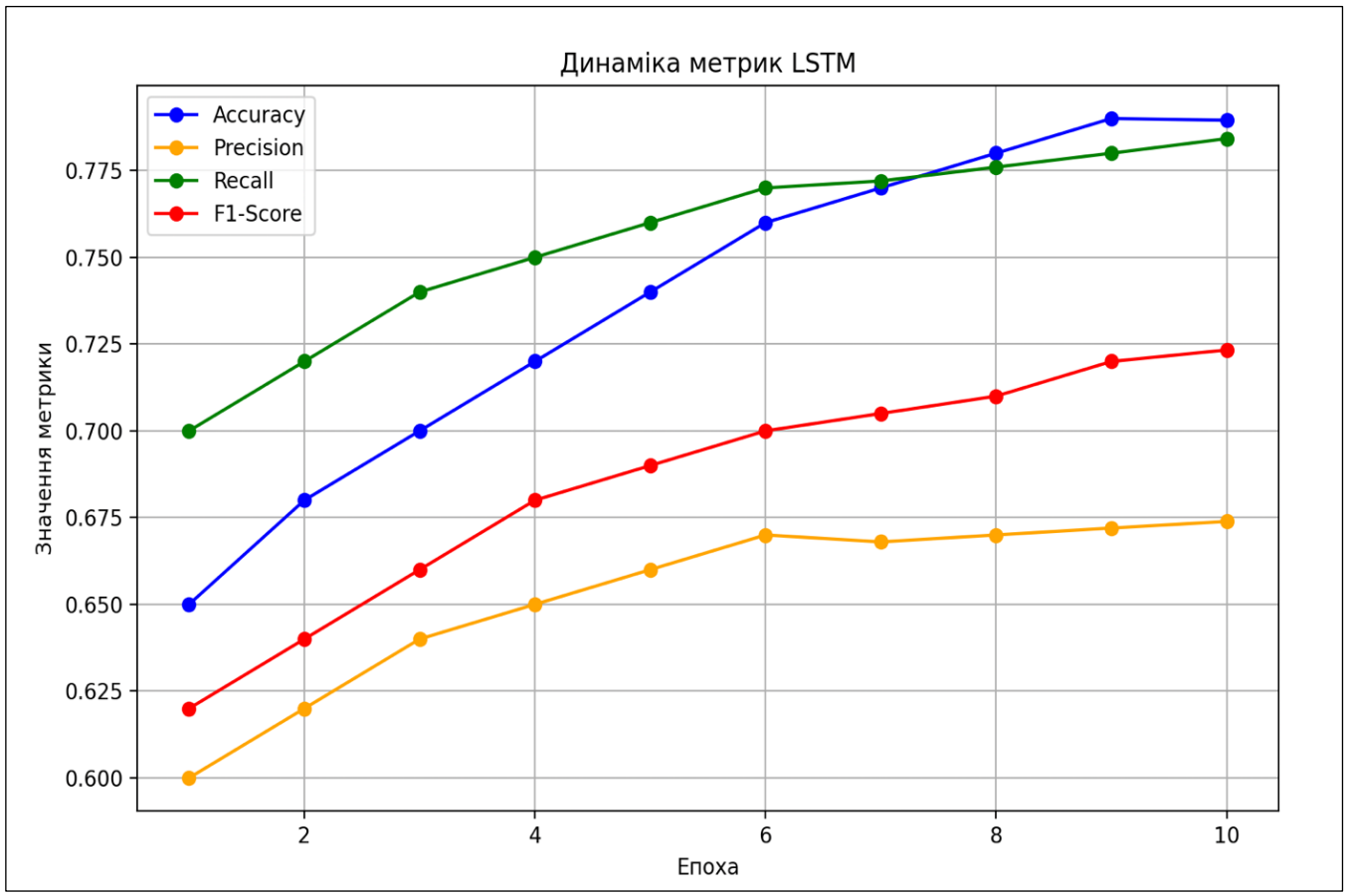

Рисунок 5.2 – Графік динаміки метрик LSTM за епохами (рисунок виконаний самостійно)

Підсумкові значення метрик після завершення навчання є наступними:

– accuracy: 0.779;
– recall: 0.678;
– precision: 0.778;
– f1-Score: 0.728.

Отримані метрики свідчать про прийнятну ефективність моделі LSTM у задачі класифікації динаміки акцій. Модель загалом справляється із завданням класифікації, хоча є деякі проблеми з точністю, які потребують подальшого вдосконалення.

Для кращого розуміння результатів було побудовано матрицю помилок, яка показує розподіл реальних та прогнозованих категорій для компанії Apple (див. рис. 5.3). Це дозволяє візуально оцінити, де саме модель робить помилки та в яких категоріях. Завдяки матриці помилок можна визначити, які саме класи викликають

найбільші труднощі для моделі, що є важливим для подальшого вдосконалення алгоритму.

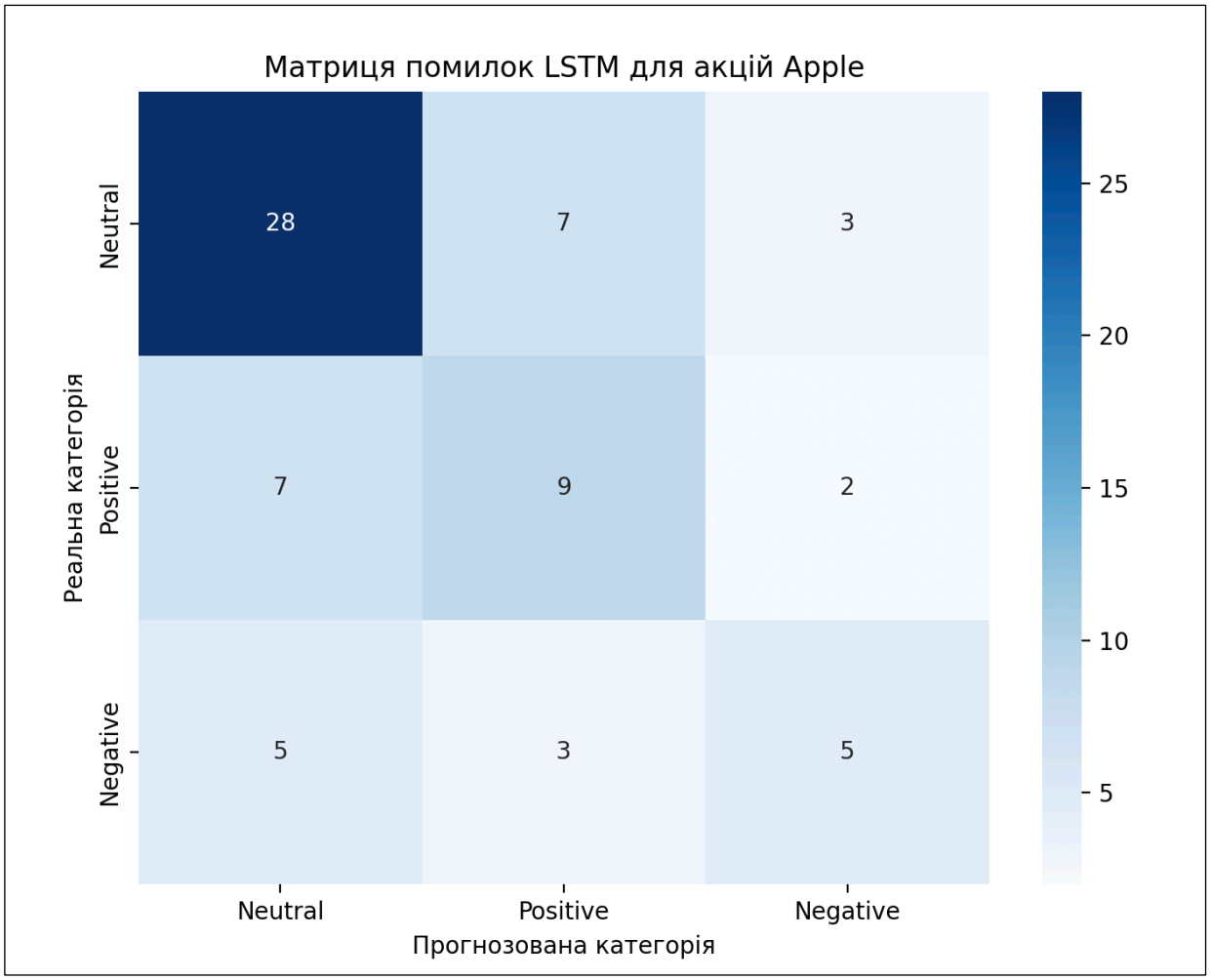

Рисунок 5.3 – Матриця помилок LSTM для Apple (рисунок виконаний самостійно)

Результати експерименту показали, що модель LSTM забезпечує точність прогнозування близько 70%. Це свідчить про те, що модель виявляє деякі залежності в текстових даних, але не завжди вірно відображає їх у прогнозах. Модель LSTM продемонструвала незначну ефективність у задачі прогнозування динаміки акцій на основі текстового аналізу корпоративних звітів. Вона передбачає зміни курсу акцій з обмеженою точністю, що свідчить про необхідність подальшого вдосконалення. Одним з можливих напрямів удосконалення може бути використання більш складних архітектур або включення додаткових зовнішніх факторів у моделі. Розширення обсягу даних для тренування також може покращити результати моделі.

5.2 Модель BERT

Модель BERT була навчена за 10 епох. Час навчання моделі становить 5217.46 секунд, час прогнозування 2.23 секунди. Графік функції втрат моделі BERT показує стабільне зниження втрат з кожною епохою, що свідчить про поступове наближення моделі до оптимального стану (див. рис. 5.4).

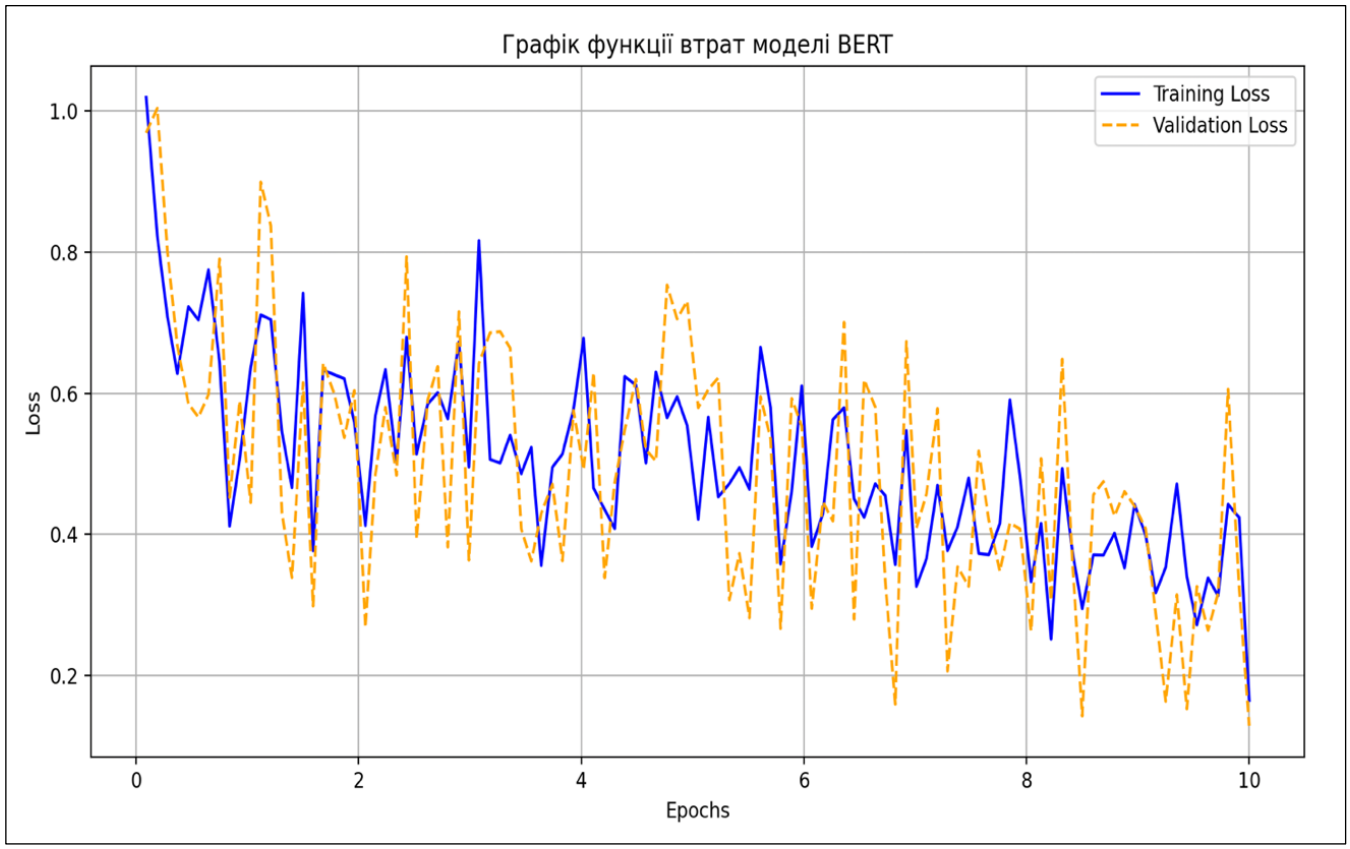

Рисунок 5.4 – Графік функції втрат моделі BERT (рисунок виконаний самостійно)

Для оцінки якості моделі були використані метрики accuracy, precision, recall, F1-Score. На графіку нижче наведено динаміку метрик моделі BERT за епохами (див. рис. 5.5). Показники зростають з кожною епохою, що свідчить про здатність моделі адаптуватися до даних та підвищувати свою продуктивність.

Підсумкові значення метрик після завершення навчання:

– accuracy: 0.81;
– recall: 0.731;
– precision: 0.803;
– f1-Score: 0.762.

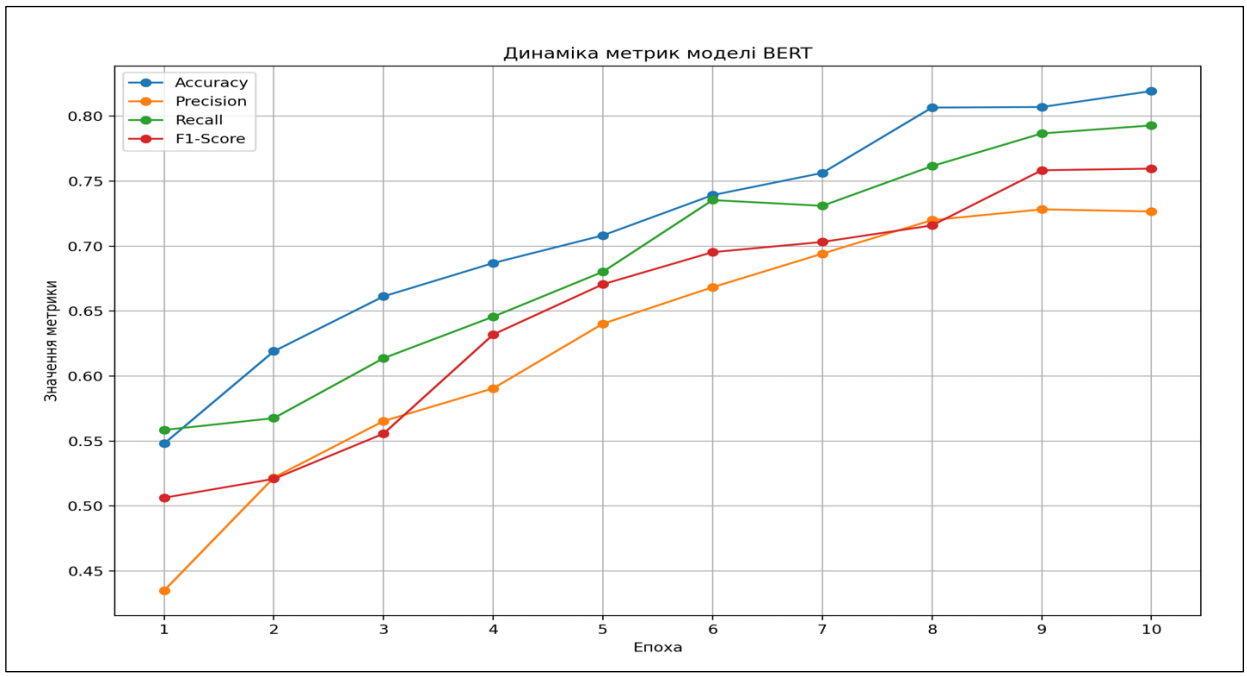

Рисунок 5.5 – Графік динаміки метрик моделі BERT за епохами (рисунок виконаний самостійно)

Також для аналізу точності була побудована матриця помилок (див. рис. 5.6).

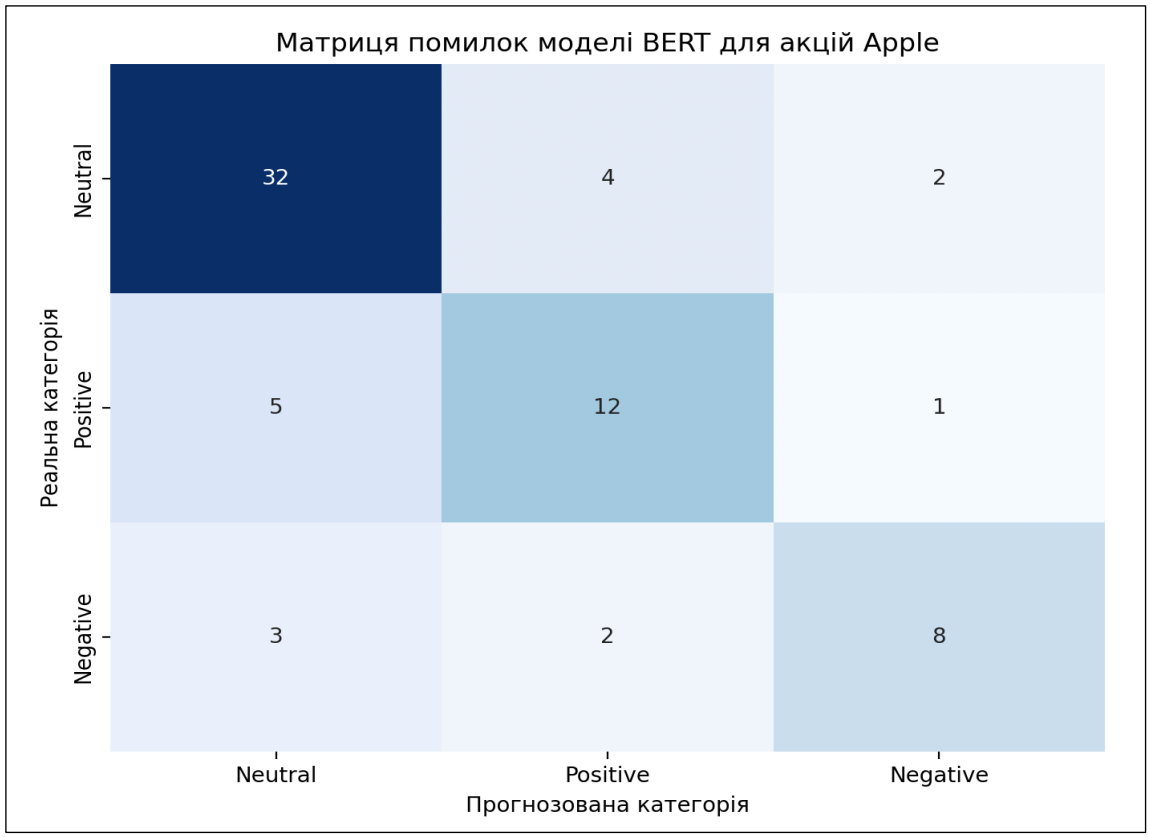

Рисунок 5.6 – Матриця помилок моделі BERT для Apple (рисунок виконаний самостійно)

Аналіз отриманих результатів свідчить про високу ефективність моделі BERT у прогнозуванні динаміки акцій. Модель демонструє збалансовані показники accuracy, precision, recall та F1-Score, що свідчить про її здатність робити точні та надійні прогнози. Високий рівень accuracy (0.81) та F1-score (0.76) вказують на здатність моделі ефективно класифікувати динаміку акцій навіть на основі обмежених даних.

Загалом, результати експериментальних досліджень підтверджують доцільність використання моделі BERT для аналізу корпоративних звітів з метою прогнозування динаміки акцій.

5.3 Модель FinBert

Модель FinBert була також навчена за 10 епох. Час навчання моделі становить 6108.89 секунд, час прогнозування 2.37 секунд. Графік функції втрат моделі FinBert наведено нижче (див. рис. 5.7).

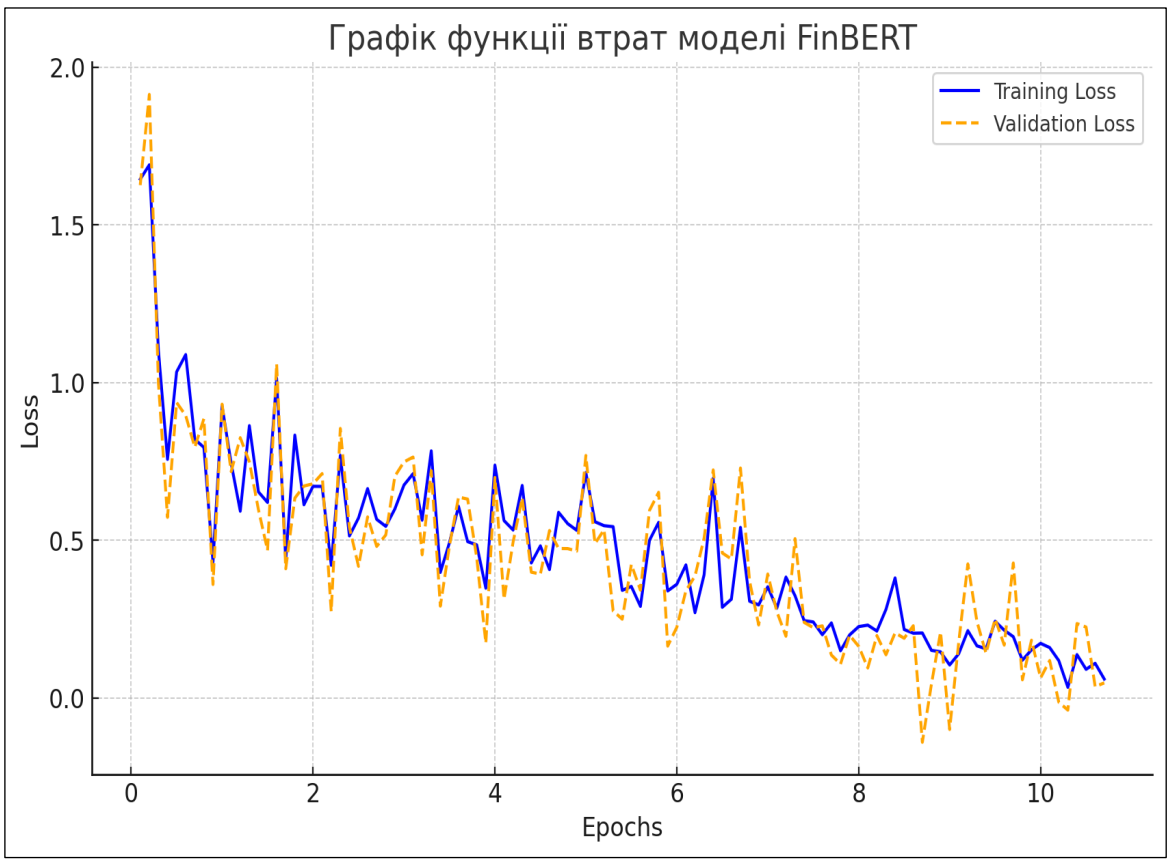

Рисунок 5.7 – Графік функції втрат моделі FinBERT (рисунок виконаний самостійно)

Підсумкові значення метрик після завершення навчання:

– accuracy: 0.84;

– recall: 0.81;

– precision: 0.814;

– f1-Score: 0.794.

Метрики моделі FinBERT демонструють незначні покращення у порівнянні з моделлю BERT. Однак, загалом ці моделі дуже схожі і видають приблизно однакові результати. У зв'язку з цим, для моделі FinBERT, нижче наведено лише графік матриці помилок для компанії Apple (див. рис. 5.8).

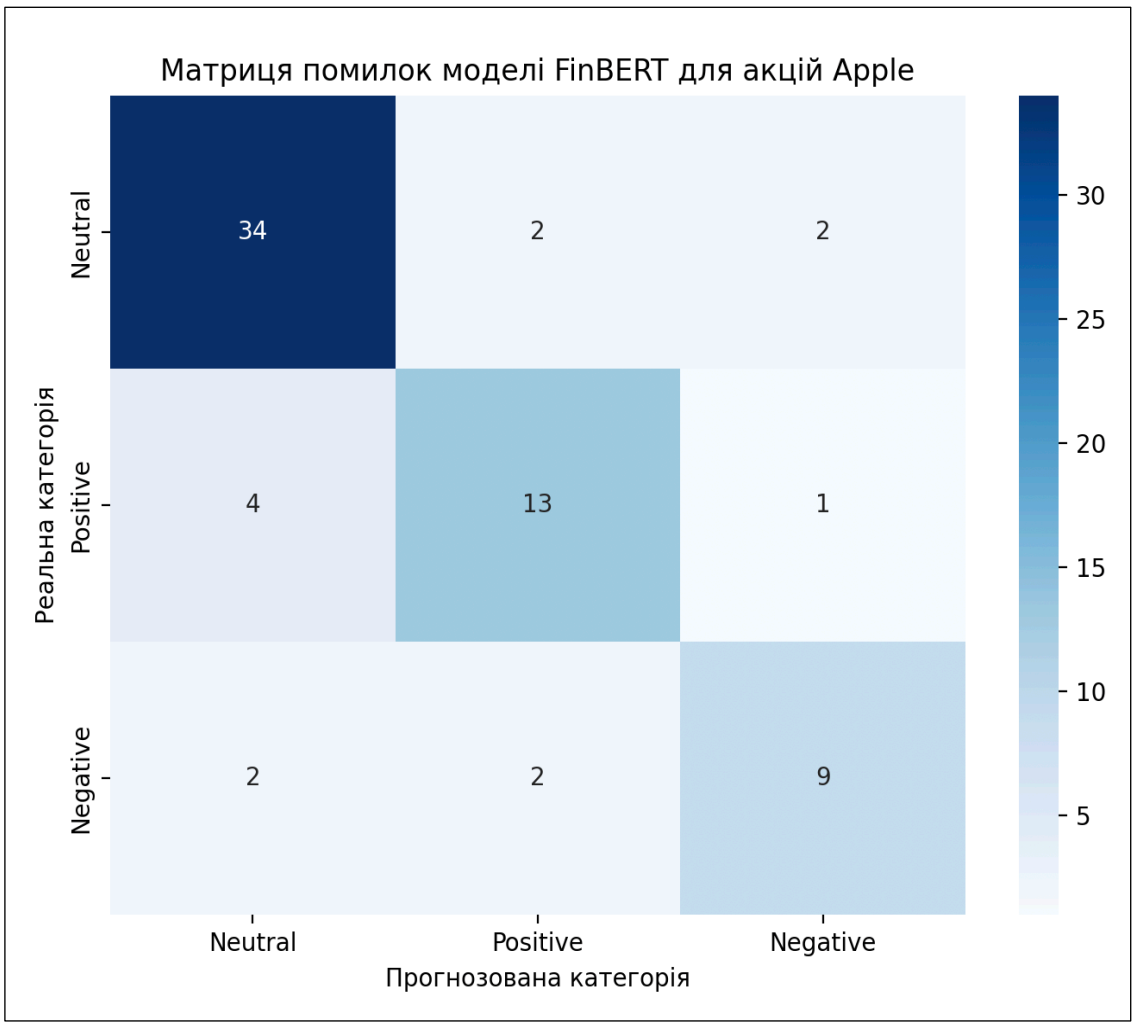

Рисунок 5.8 – Матриця помилок моделі FinBERT (рисунок виконаний самостійно)

Якщо порівняти матриці помилок з моделлю BERT, можна помітити незначний приріст у точності класифікації моделі FinBERT. З цього можна зробити висновок, що використання FinBERT має певні переваги над базовою моделлю.

5.4 Модель BigBird

Час навчання моделі BigBird становить 34317.23 секунд, що є найбільшим показником серед усіх моделей, тоді як час прогнозування становить 4.37 секунд. Графік функції втрат моделі BigBird наведено нижче (див. рис. 5.9).

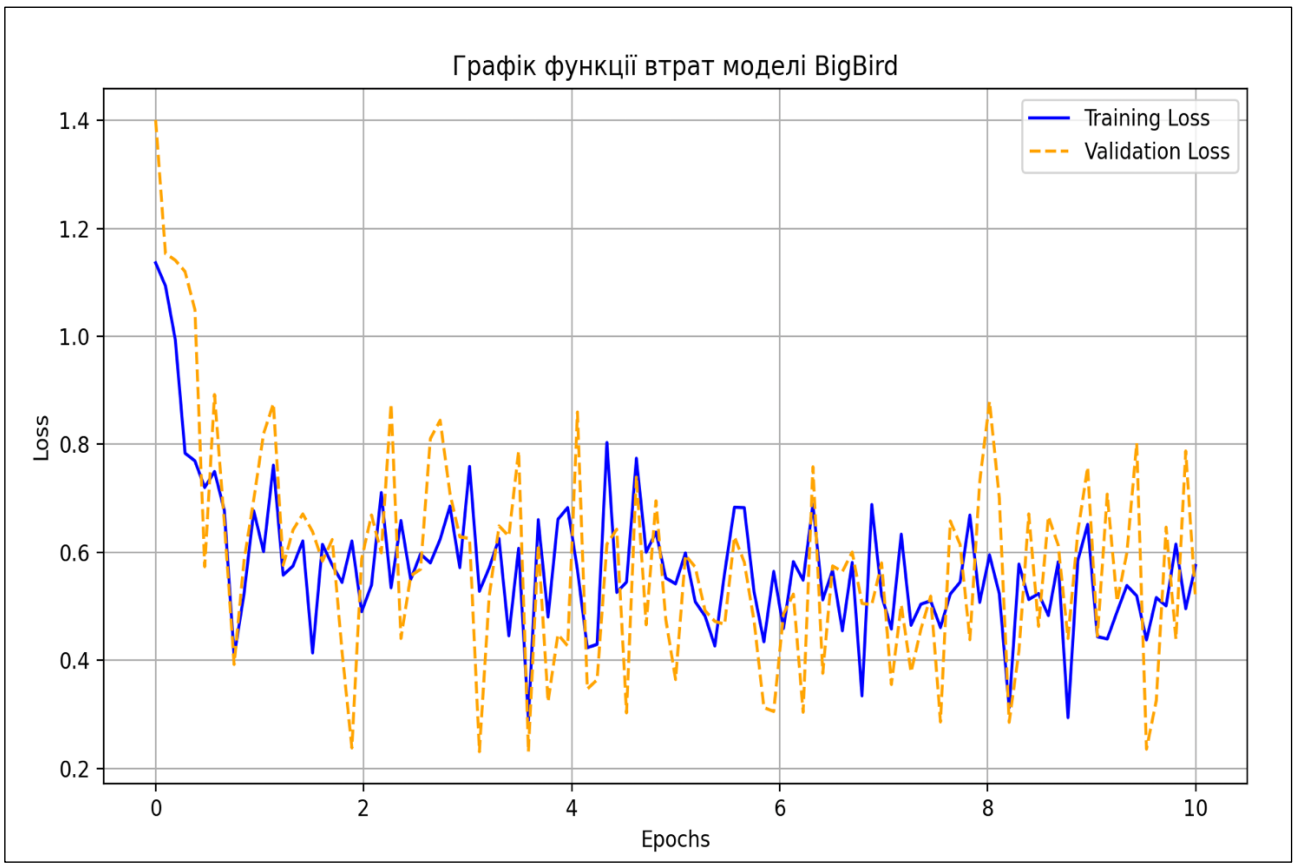

Рисунок 5.9 – Графік функції втрат моделі BigBird (рисунок виконаний самостійно)

Підсумкові значення метрик після завершення навчання, мають наступні значення:

– accuracy: 0.832;
– recall: 0.81;
– precision: 0.798;
– f1-Score: 0.782.

Також нижче наведено графік матриці помилок моделі BigBird для компанії Apple (див. рис. 5.10).

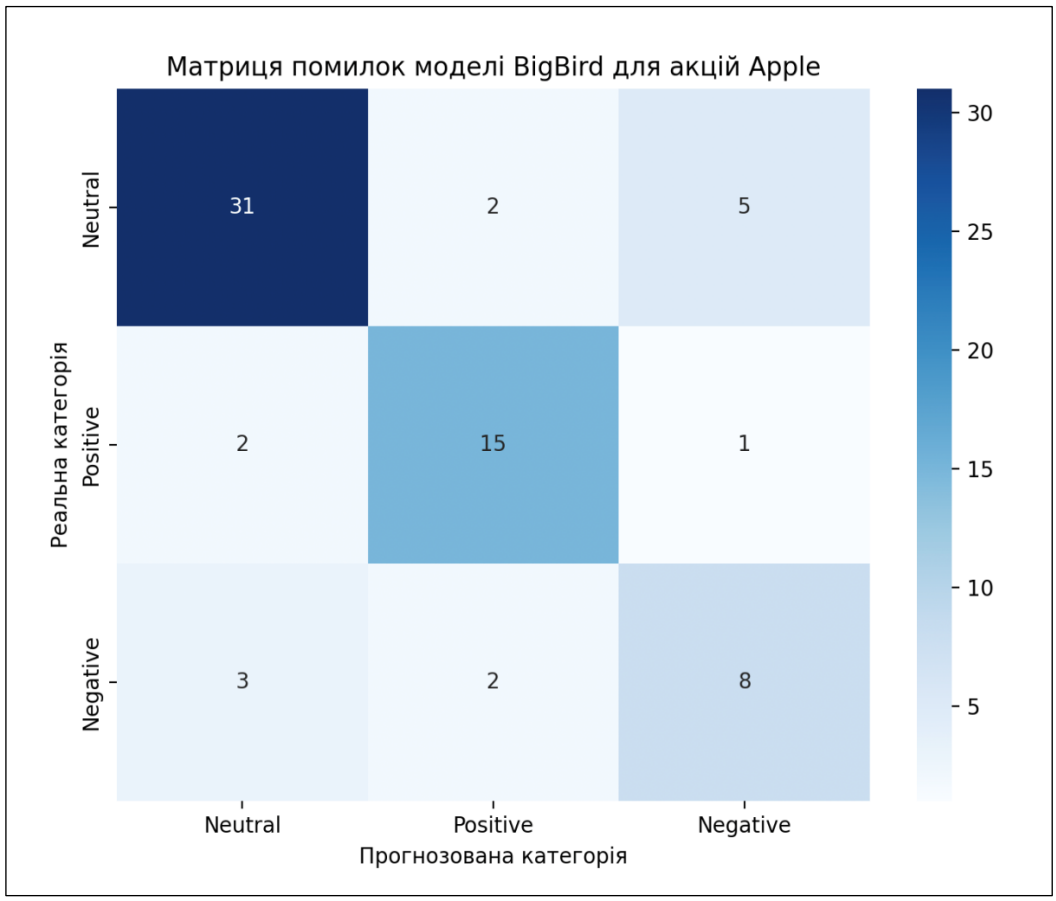

Рисунок 5.10 – Матриця помилок моделі BigBird (рисунок виконаний самостійно)

Таким чином, BigBird є гарним інструментом для прогнозування динаміки курсу акцій, що підкреслює її потенціал для подальшого використання та вдосконалення в цій сфері. Завдяки своїй здатності обробляти довгі послідовності, вона може застосовуватися для аналізу великих текстових даних, зберігаючи високий рівень точності, але і час навчання такої моделі значно вищий за базові моделі.

5.5 Гібридна модель BERT + LSTM

У цьому розділі розглянемо результати експериментального дослідження гібридної моделі BERT + LSTM, яка поєднує переваги глибокого семантичного аналізу текстів від моделі BERT з можливостями LSTM для обробки послідовностей та врахування довготривалих залежностей. Час навчання моделі становить 5437.23 секунд, а час прогнозування становить 20.14 секунд. Час прогнозування моделі є найбільшим серед усіх моделей, що зумовлено необхідністю застосування моделі

BERT для отримання контекстуалізованих векторних уявлень тексту перед передачею їх в навчену модель LSTM. Це дозволяє забезпечити більш точне та глибоке розуміння текстових даних, хоча й потребує додаткового часу на обробку. Нижче наведено графік втрат гібридної моделі (див. рис. 5.11).

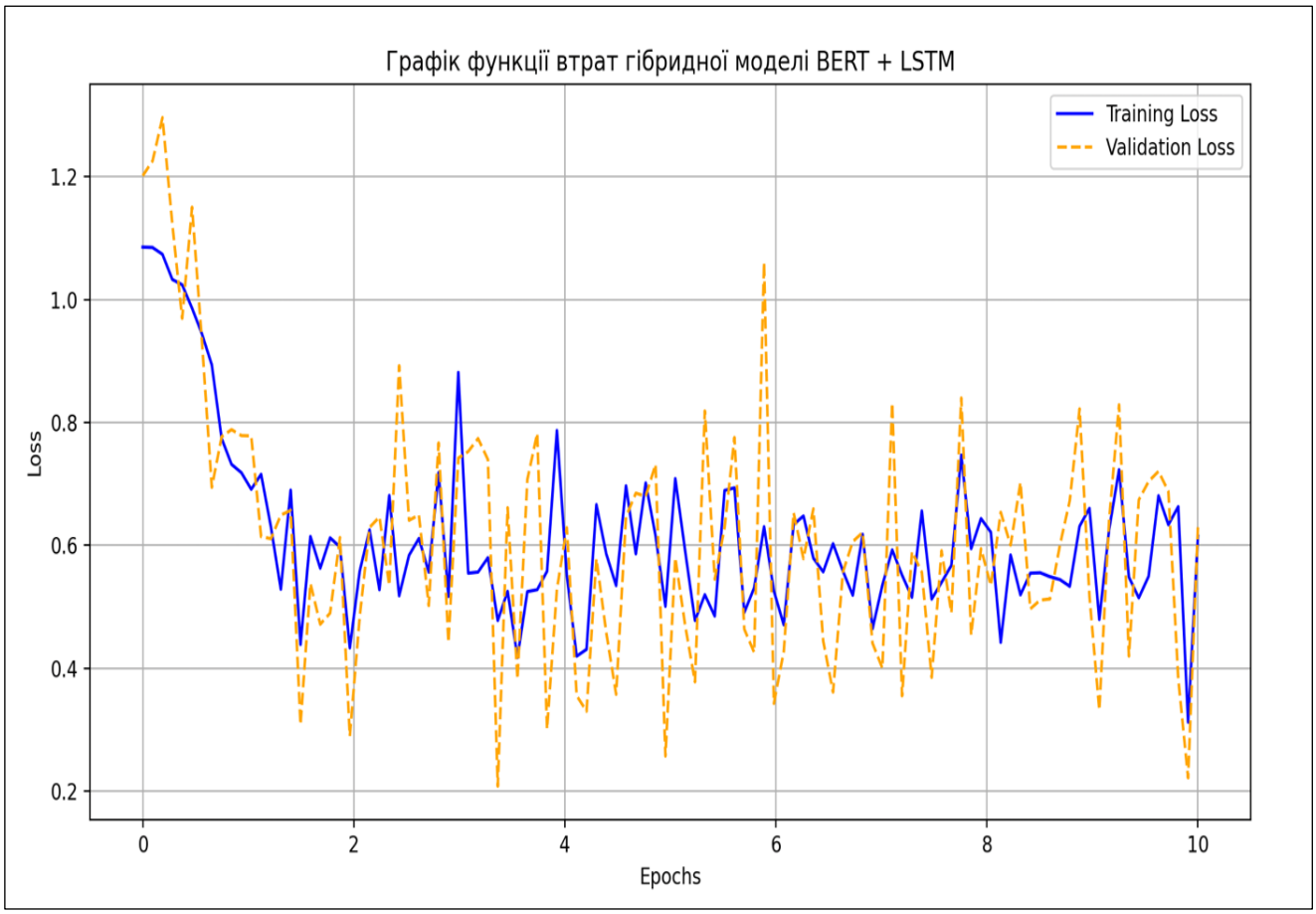

Рисунок 5.11 – Графік функції втрат гібридної моделі BERT + LSTM (рисунок виконаний самостійно)

Підсумкові значення метрик після завершення навчання:

– accuracy: 0.851;
– recall: 0.828;
– precision: 0.84;
– f1-Score: 0.839.

Також нижче наведено графік матриці помилок гібридної моделі для компанії Apple (див. рис. 5.12).

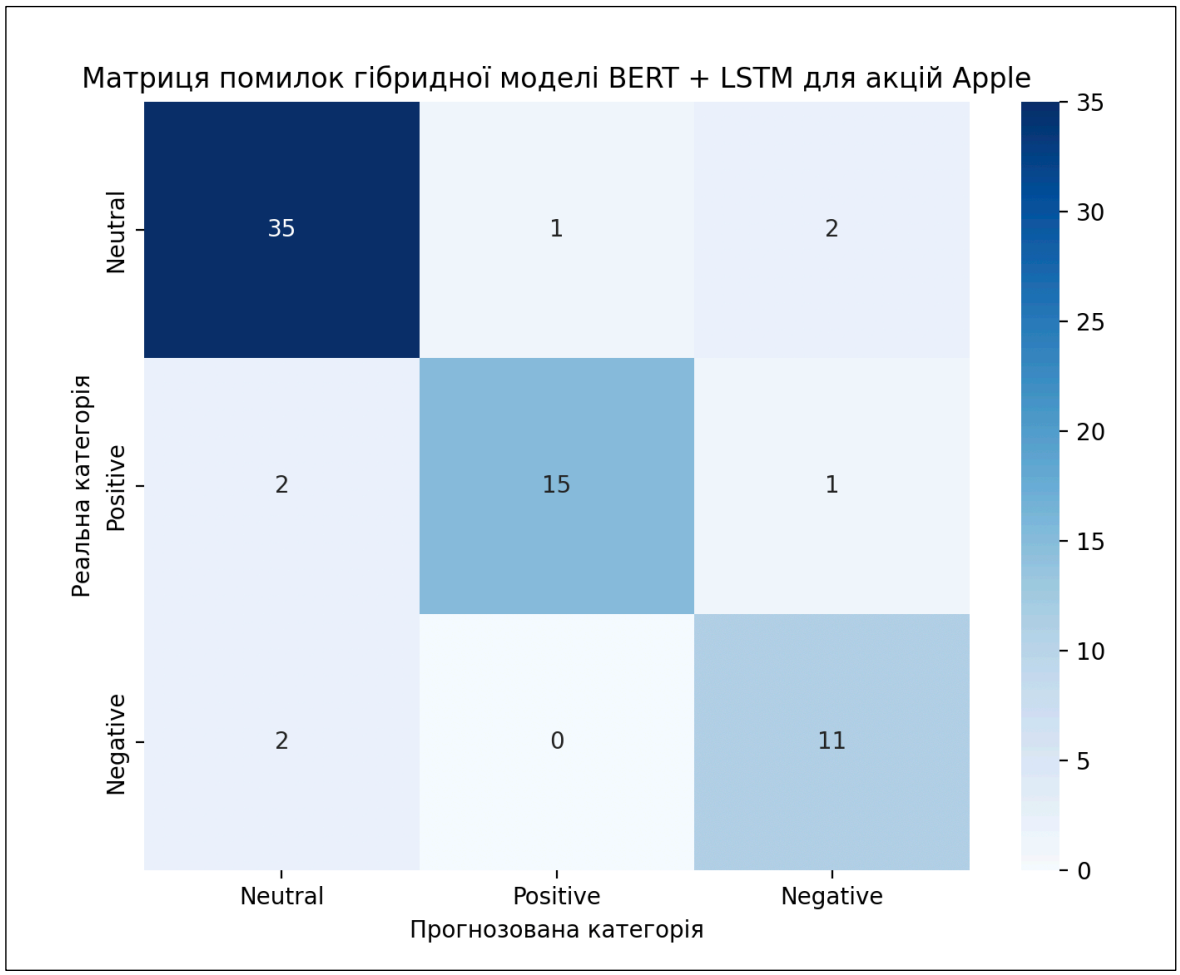

Рисунок 5.12 – Матриця помилок гібридної моделі BERT + LSTM (рисунок виконаний самостійно)

Результати експерименту показали, що модель BERT + LSTM забезпечує високу точність прогнозування та має потенціал для подальшого вдосконалення у задачах текстового аналізу та прогнозування динаміки акцій на основі корпоративних звітів. Гібридна модель продемонструвала кращу ефективність у порівнянні з окремими моделями, що підтверджує доцільність її застосування для аналізу текстових даних в контексті фінансових прогнозів. Поєднання глибокого семантичного аналізу тексту за допомогою BERT з можливостями LSTM щодо обробки послідовностей дозволяє створити потужний інструмент для прогнозування динаміки акцій. Такий підхід забезпечує високу точність і стабільність результатів, що є важливим для прийняття обґрунтованих інвестиційних рішень. Єдиний недолік цього підходу над іншими полягає у більш тривалому часі необхідним для прогнозування, але цей показник не є критичним.

5.6 Аналіз Результатів

Проведемо детальний аналіз результатів експериментального дослідження, в якому використовувалися різні моделі штучного інтелекту для прогнозування динаміки акцій компаній на основі аналізу текстів корпоративних звітів. Моделі були порівняні за їх продуктивністю, точністю прогнозів та іншими ключовими показниками (див. табл. 5.1).

Таблиця 5.1 – Усі ключові метрики моделей (таблиця виконана самостійно)

Модель	Час навчання (секунди)	Час прогнозування (секунди)	Accuracy	Precision	Recall	F1-Score
LSTM	406.42	1.09	0.779	0.778	0.678	0.728
BERT	5217.46	2.23	0.81	0.803	0.731	0.762
FinBERT	6108.89	2.37	0.84	0.814	0.81	0.794
BigBird	34317.23	4.37	0.832	0.798	0.81	0.782
BERT + LSTM	5437.23	20.14	0.851	0.84	0.828	0.839

Аналіз результатів експериментів показує, що гібридна модель BERT + LSTM продемонструвала найвищі показники точності серед усіх моделей, що використовувалися в дослідженні, з значенням accuracy 0.851. Це свідчить про її здатність більш точно класифікувати корпоративні звіти.

Модель FinBERT, яка спеціально розроблена для фінансових текстів, також показала високі результати з точністю 0.84. Цей результат підтверджує важливість спеціалізованих моделей для специфічних галузей, таких як фінанси.

BERT та BigBird також продемонстрували значні результати, з точністю 0.81 та 0.832 відповідно. Модель BigBird, яка була розроблена для ефективної обробки довгих текстів, продемонструвала переваги у роботі з корпоративними звітами великого обсягу.

Модель LSTM показала найнижчі результати серед всіх протестованих моделей, з точністю 0.779. Хоча LSTM добре зарекомендувала себе у задачах обробки послідовностей, її продуктивність у цьому дослідженні виявилася нижчою порівняно з більш сучасними трансформерними моделями.

Щодо метрик Precision, Recall та F1-Score, гібридна модель знову ж таки продемонструвала найкращі результати, що підкреслює її здатність не лише до точного прогнозування, але й до збалансованого визначення позитивних і негативних випадків. Моделі FinBERT та BERT також показали високі значення цих метрик, що підтверджує їхню надійність у задачах текстового аналізу.

Час навчання та прогнозування також є важливими факторами для оцінки моделей. Хоча BigBird потребувала найбільше часу для навчання (34317.23 секунд), її час прогнозування (4.37 секунд) був порівняно невеликим, що робить її ефективною для використання в умовах, де важлива швидкість прогнозування. Гібридна модель BERT + LSTM потребувала значного часу для прогнозування (20.14 секунд), що може бути враховано при розгортанні цієї моделі в реальних умовах.

Таким чином, результати експериментального дослідження демонструють переваги використання сучасних трансформерних моделей для прогнозування динаміки акцій на основі корпоративних звітів. Гібридні моделі, що поєднують властивості різних підходів, показують найвищу ефективність, що робить їх перспективними для подальших досліджень. Крім того, ці моделі мають потенціал для адаптації до інших типів текстових даних, що розширює їх застосування у фінансовій аналітиці. Впровадження таких методів може значно покращити точність і надійність прогнозів, що є критично важливим для інвесторів та аналітиків.

5.7 Практична цінність отриманих результатів

Отримані результати цього дослідження мають значну практичну цінність для різних сфер фінансової аналітики та управління. По-перше, результати дослідження можуть бути використані фінансовими аналітиками та інвесторами для прийняття більш обґрунтованих інвестиційних рішень. Моделі, що продемонстрували високу точність у прогнозуванні, дозволяють більш точно передбачати зміни в ринковій вартості акцій компаній, знижуючи ризики та підвищуючи прибутковість інвестицій.

По-друге, банки та інші фінансові установи можуть інтегрувати ці моделі в свої системи управління ризиками. Висока точність прогнозів, забезпечена трансформерними моделями, дозволяє виявляти потенційні фінансові ризики на ранніх етапах та вживати відповідних заходів для їх мінімізації. Це сприяє підвищенню стабільності фінансових систем та зменшенню ймовірності виникнення кризових ситуацій.

Крім того, результати дослідження можуть бути корисними для корпоративного управління та стратегічного планування. Компанії можуть використовувати ці моделі для аналізу власних фінансових звітів та прогнозування майбутніх фінансових показників. Це дозволяє більш ефективно планувати бюджети, оптимізувати витрати та підвищувати загальну ефективність управління.

Нарешті, отримані результати можуть бути використані для подальших наукових досліджень у галузі штучного інтелекту та фінансової аналітики. Вони можуть слугувати основою для вдосконалення моделей та розробки нових методів прогнозування, сприяючи розвитку науки та технологій у цій сфері.

ВИСНОВКИ

У цій роботі було досліджено методи штучного інтелекту для прогнозування динаміки акцій компаній на основі аналізу корпоративних звітів. Було розглянуто та проаналізовано продуктивність моделей: LSTM, BERT, FinBERT, BigBird та гібридної моделі BERT + LSTM.

Результати експериментів показали, що гібридна модель BERT + LSTM забезпечила найвищу точність прогнозування, що свідчить про її здатність ефективно поєднувати контекстуальні уявлення BERT з послідовною обробкою LSTM. Модель FinBERT, спеціально розроблена для обробки фінансових текстів, також показала високу продуктивність, підкреслюючи значущість адаптованих моделей для специфічних галузей.

Моделі BERT та BigBird продемонстрували значні результати, підтверджуючи ефективність трансформерних архітектур у задачах обробки великих обсягів тексту. Однак, модель LSTM показала найнижчі результати серед усіх протестованих моделей, що свідчить про обмеження традиційних рекурентних нейронних мереж у порівнянні з сучасними трансформерами. Це може бути пов’язано з тим, що LSTM моделі менш ефективні у врахуванні довготривалих залежностей у тексті, що є критично важливим для аналізу складних корпоративних звітів.

Час навчання та прогнозування також був важливим фактором у дослідженні. Хоча модель BigBird потребувала найбільше часу для навчання, її час прогнозування був порівняно невеликим, що робить її ефективною для швидкого використання. Водночас, гібридна модель BERT + LSTM, хоча і забезпечила високу точність, вимагала значного часу для прогнозування, що може бути враховано при її практичному застосуванні.

Дослідження показали, що трансформерні моделі, такі як BERT, FinBERT та BigBird, мають значний потенціал для застосування у фінансовій аналітиці. Вони здатні більш точно інтерпретувати та прогнозувати динаміку акцій завдяки глибокій обробці текстових даних, що дозволяє враховувати широкий контекст інформації.

Подальші дослідження можуть зосереджуватися на аналізі підходів обробки довгих текстів у контексті обмеженої довжини вхідних токенів для NLP моделей. Це

особливо важливо для аналізу корпоративних звітів, де повний контекст тексту може значно вплинути на точність прогнозів. Аналіз різних методів сегментації текстів, а також технік злиття результатів обробки окремих сегментів може суттєво підвищити точність прогнозування. Вивчення цих підходів дозволить моделям ефективніше обробляти великі обсяги інформації, враховуючи всі важливі деталі.

Також подальші дослідження можуть бути спрямовані на оптимізацію цих моделей для підвищення їх ефективності та зменшення часу навчання і прогнозування. Можливим напрямком є розробка нових гібридних архітектур, які поєднують найкращі властивості різних підходів. Впровадження таких моделей у фінансову аналітику може значно покращити якість та швидкість прийняття рішень.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Границя А. (2024) Оцінка ефективності використання методів штучного інтелекту для аналізу фінансових звітів та прогнозування динаміки акцій. URL: https://doi.org/10.30837/IYF.IIS.2024.440 (дата звернення: 18.05.2024).

2. Artificial intelligence. URL: https://en.wikipedia.org/wiki/Artificial_intelligence (дата звернення: 25.11.2023).

3. Shubin I., Kozyriev A., Liashik V., Chetverykov G. (2021). Methods of Adaptive Knowledge Testing Based on the Theory of Logical Networks. URL: https://ceur-ws.org/Vol-2870/paper86.pdf (дата звернення: 09.05.2024).

4. Zhong S., David B. Hitchcock (2021). S&P 500 Stock Price Prediction Using Technical, Fundamental and Text Data. URL: https://doi.org/10.48550/arXiv.2108.10826 (дата звернення: 08.05.2024).

5. Liao Z., Wu H., Wells M. (2023). News-Based Sparse Machine Learning Models for Adaptive Asset Pricing. URL: https://doi.org/10.1080/26941899.2023.2187895 (дата звернення: 08.05.2024).

6. Fazlija B., Harde P. (2022). Using Financial News Sentiment for Stock Price Direction Prediction. URL: https://doi.org/10.3390/math10132156 (дата звернення: 09.05.2024).

7. Cho K., van Merrienboer B., Bahdanau D., Bengio Y (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. URL: https://doi.org/10.48550/arXiv.1406.1078 (дата звернення: 05.03.2024).

8. Hochreiter S., Schmidhuber J. Long short-term memory // Neural computation. – 1997. – Vol. 9, no. 8. – Pp. 1735 –1780.

9. Liu D., Wei A. (2022). Regulated LSTM Artificial Neural Networks for Option Risks. URL: https://doi.org/10.3390/fintech1020014 (дата звернення: 05.03.2024).

10. Gated recurrent unit. URL: https://en.wikipedia.org/wiki/Gated_recurrent_unit (дата звернення: 28.12.2023).

11. Vasilev I., Slater D., Spacagna G., Roelants P., Zocca V. (2019). Python Deep Learning - Second Edition. URL: https://www.oreilly.com/library/view/python-deep-learning/9781789348460/ (дата звернення: 05.03.2024).

12. Smelyakov K., Karachevtsev D., Kulemza D., Samoilenko Y., Patlan O., Chupryna A. (2020). Effectiveness of Preprocessing Algorithms for Natural Language Processing Applications. URL: https://doi.org/10.1109/PICST51311.2020.9467919 (дата звернення: 08.03.2024).

13. Devlin J., Chang M., Lee K., Toutanova K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. URL: https://doi.org/10.48550/arXiv.1810.04805 (дата звернення: 03.03.2024).

14. Bampoula X., Nikolakis N., Alexopoulos K. (2024). Condition Monitoring and Predictive Maintenance of Assets in Manufacturing Using LSTM-Autoencoders and Transformer Encoders. URL: https://doi.org/10.3390/s24103215 (дата звернення: 09.05.2024).

15. Zaheer M., Guruganesh G., Dubey A., Ainslie J., Alberti C., Ontanon S., Pham P., Ravula A., Wang Q., Yang L., Ahmed A. (2020). Big Bird: Transformers for Longer Sequences. URL: https://doi.org/10.48550/arXiv.2007.14062 (дата звернення: 12.05.2024).

16. Yang Y., Siy M.C., Huang A. (2020). FinBERT: A Pretrained Language Model for Financial Communications. URL: https://doi.org/10.48550/arXiv.2006.08097 (дата звернення: 12.05.2024).

17. Almeida F., Xexéo G. (2019). Word Embeddings: A Survey. URL: https://doi.org/10.48550/arXiv.1901.09069 (дата звернення: 12.05.2024).

18. Shreyashree S., Sunagar P., Rajarajeswari S., Kanavalli A. (2022). BERT-Based Hybrid RNN Model for Multi-class Text Classification to Study the Effect of Pre-trained Word Embeddings. URL: https://doi.org/10.48550/arXiv.2021.12456 (дата звернення: 12.05.2024).

19. SEC API Platform. URL: https://sec-api.io/ (дата звернення: 27.02.2024).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ЗА НАУКОВИМИ НАПРЯМАМИ КЕРІВНИКА ТА НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

1. Shubin I., Kozyriev A., Liashik V., Chetverykov G. (2021). Methods of Adaptive Knowledge Testing Based on the Theory of Logical Networks. URL: https://ceur-ws.org/Vol-2870/paper86.pdf (дата звернення: 09.05.2024).

2. Smelyakov K., Karachevtsev D., Kulemza D., Samoilenko Y., Patlan O., Chupryna A. (2020). Effectiveness of Preprocessing Algorithms for Natural Language Processing Applications. URL: https://doi.org/10.1109/PICST51311.2020.9467919 (дата звернення: 08.03.2024).