НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Навчально-науковий інститут прикладного системного аналізу
Кафедра штучного інтелекту

«На правах рукопису»
УДК004.891:00485]:615.015.2(043.3)

До захисту допущено:
В. о. завідувачки кафедри
_____ Ірина ДЖИГИРЕЙ
«___»_____20__ р.

Магістерська дисертація

на здобуття ступеня магістра
за освітньо-професійною програмою «Системи і методи штучного інтелекту»
зі спеціальності 122 «Комп'ютерні науки»
на тему: « Інтелектуальна система визначення взаємодії між ліками **»**

Виконав:
студент ІІ курсу, групи КІ-41мп
Мацуєв Роман Олександрович _____

Науковий керівник:
професор кафедри ШІ, д.т.н., професор,
Синєглазов Віктор Михайлович _____

Рецензент:
завідувач кафедри ІБ ННФТІ, д.т.н., професор,
Ланде Дмитро Володимирович _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент (-ка) _____

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Навчально-науковий інститут прикладного системного аналізу

Кафедра штучного інтелекту

Рівень вищої освіти – другий (магістерський)
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма «Системи і методи штучного інтелекту»

ЗАТВЕРДЖУЮ
В. о. завідувачки кафедри
_____ Ірина ДЖИГИРЕЙ
«__» ____ 2025 р.

ЗАВДАННЯ
на магістерську дисертацію студенту

1. Тема дисертації «Інтелектуальна система визначення взаємодії між ліками», науковий керівник дисертації Синєглазов Віктор Михайлович, завідувач кафедри ІБ ННФТІ, д.т.н., професор, затверджені наказом по ННІПСА від "06" листопада 2025 р. № 4837-с.

2. Термін подання студентом дисертації «12» грудня 2025 року _____

3. Об’єкт дослідження Інтелектуальна система визначення взаємодії між ліками

4. Вихідні дані Перелік вірогідних ефектів від дії ліків на організм пацієнта

5. Перелік завдань, які потрібно розробити: 1. Аналіз предметної області 2.Виконати огляд літературних джерел 3.Вибір технології для розробки 4. Виконати збір даних 5. Розробити стратегію препроцесингу даних 6. Аналіз існуючих рішень 7. Проектування архітектури моделі 8.Аналіз результатів

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: Рисунки, які відображають архітектуру моделі, та результати

7. Орієнтовний перелік публікацій: тези в збірці матеріалів IV ВНПК «Системні науки та інформатика»

8. Дата видачі завдання «29» серпня 2025року

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз можливих тем	01.09.2025 – 14.09.2025	Виконано
2	Затвердження теми роботи	20.09.2025	Виконано
3	Аналіз предметної області	21.09.2025 – 28.09.2025	Виконано
4	Аналіз існуючих рішень	29.09.2025 – 10.10.2025	Виконано
5	Проєктування і реалізація моделі	13.10-2025 – 31.11.2025	Виконано
6	Аналіз результатів	01.12.2025 – 07.12.2025	Виконано
7	Оформлення пояснювальної записки	08.12.2025 – 12.12.2025	Виконано

Студент Роман МАЦУЄВ
Науковий керівник Віктор СИНЄГЛАЗОВ,

РЕФЕРАТ

ПРОГНОЗУВАННЯ ВЗАЄМОДІЇ ЛІКІВ, DDI, ТРАНСФОРМЕР-ЕНКОДЕР, МУЛЬТИМОДАЛЬНЕ НАВЧАННЯ, SELF-ATTENTION, K-FOLD CROSS-VALIDATION, БАГАТОМІТКОВА КЛАСИФІКАЦІЯ, PYTORCH

Структура та обсяг роботи. Робота складається з трьох розділів, містить 94 сторінки, 10 рисунків, 17 таблиць, 45 посилань.

Актуальність теми. Прогнозування взаємодії лікарських засобів (DDI) є актуальним завданням фармакології та клінічної медицини через ризик побічних реакцій і зниження ефективності лікування. Зростання поліпрогмазії зумовлює потребу в автоматизованих високоточних методах прогнозування. Наявні підходи, орієнтовані переважно на одномодальні ознаки, не забезпечують ефективного моделювання складних нелінійних взаємозв’язків між гетерогенними фармакологічними та хімічними даними.

Мета роботи. Підвищення точності багатоміткового прогнозування подій DDI шляхом розробки моделі DDI-TransMDL на основі архітектури Трансформер-Енкодера для інтеграції мультимодальних ознак.

Методи дослідження. Застосовано методи системного аналізу та глибокого навчання, зокрема Трансформер-Енкодер з механізмом Multi-Head Self-Attention. Реалізовано інжиніринг ознак із включенням коефіцієнта Жаккарда та оцінювання моделі за допомогою 5-Fold Cross-Validation і багатоміткових метрик Micro/Macro F1-Score, ROC-AUC та AUPR.

Наукова новизна. Запропоновано архітектуру DDI-TransMDL для інтеграції чотирьох гетерогенних модальностей (SMILES, Target, Enzyme, Pathway) у задачі прогнозування DDI, а також метод включення коефіцієнта Жаккарда у вхідний вектор ознак.

Практична цінність. Модель демонструє високу прогностичну ефективність (Micro ROC-AUC ≈ 0.9893, Micro AUPR ≈ 0.8351) та може використовуватися для багатоміткового прогнозування 65 подій DDI у фармаконагляді та клінічних дослідженнях.

ABSTRACT

DRUG–DRUG INTERACTION PREDICTION, DDI, TRANSFORMER ENCODER, MULTIMODAL LEARNING, SELF-ATTENTION, K-FOLD CROSS-VALIDATION, MULTILABEL CLASSIFICATION, PYTORCH

Structure and Scope of the Thesis. The thesis consists of three chapters and comprises 94 pages, 10 figures, 17 tables, and 45 references.

Relevance of the Topic. Drug–drug interaction (DDI) prediction is a relevant task in pharmacology and clinical medicine due to the risk of adverse reactions and reduced therapeutic efficacy. The increasing prevalence of polypharmacy necessitates the development of automated and highly accurate prediction methods. Existing approaches, which are primarily based on unimodal features, fail to effectively model complex nonlinear relationships among heterogeneous pharmacological and chemical data.

Objective of the Study. The objective of this work is to improve the accuracy of multilabel DDI event prediction by developing the DDI-TransMDL model based on a Transformer Encoder architecture for the integration of multimodal features.

Research Methods. The study applies methods of system analysis and deep learning, in particular a Transformer Encoder with a Multi-Head Self-Attention mechanism. Feature engineering techniques are implemented by incorporating the Jaccard coefficient, and model evaluation is performed using 5-Fold Cross- Validation and multilabel metrics, including Micro/Macro F1-Score, ROC-AUC, and AUPR.

Scientific Novelty. A DDI-TransMDL architecture is proposed for integrating four heterogeneous modalities (SMILES, Target, Enzyme, and Pathway) in the context of DDI prediction, along with a method for incorporating the Jaccard coefficient into the input feature vector.

Practical Significance. The model demonstrates high predictive performance (Micro ROC-AUC ≈ 0.9893, Micro AUPR ≈ 0.8351) and can be used for multilabel prediction of 65 DDI events in pharmacovigilance and clinical research.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

DDI (Drug-Drug Interactions) - Взаємодія між лікарськими засобами

CNN (Convolutional Neural Networks) – Згорткова нейронна мережа

GNN (Graph Neural Networks) – Графова нейронна мережа

SVM (Support Vector Machine) – Метод опорних векторів

RF (Random Forest) – Метод випадкового лісу

CDSS (Clinical Decision Support Systems - Системи підтримки клінічних рішень

In silico - це термін, який означає дослідження або експерименти, проведені за допомогою комп'ютерних моделей або симуляцій, тобто на комп'ютері, а не в лабораторії чи живому організмі

SMILES (Simplified Molecular Input Line Entry System) - система запису хімічних структур у вигляді текстових рядків

MCNN (Multimodal Convolutional Neural Networks) - мультимодальні згорткові нейронні мережі

InChi (International Chemical Identifier) - міжнародний хімічний ідентифікатор

IUPAC (International Union of Pure and Applied Chemistry) - міжнародна недержавна організація

NLP (Natural Language Processing) – обробка природної мови

MLP (Multilayer Perceptron) - Багатошаровий перцептрон

SGD (Stochastic gradient descent) - Стохастичний градієнтний спуск

MSE (Mean squared error) – середньоквадратична похибка

MAE (Mean absolute error) – середня абсолютна похибка

BCE (Binary Cross Entropy loss) – Бінарна крос-ентропія

CCE (Categorical Cross Entropy loss) - Категорична крос-ентропія

ВСТУП

Сучасна фармакотерапія характеризується зростанням використання поліпрагмазії - одночасного призначення пацієнту п’яти і більше лікарських засобів. Хоча поліпрагмазія є неминучою при лікуванні хронічних та коморбідних захворювань, вона критично підвищує ризик виникнення взаємодії між лікарськими засобами (DDI). DDI можуть спричинити непередбачувані побічні ефекти, зниження терапевтичної ефективності препаратів, госпіталізацію та, у найгірших випадках, летальні наслідки. В умовах експоненційного зростання кількості нових хімічних сполук та лікарських комбінацій, традиційні методи виявлення DDI (наприклад, клінічні випробування або лабораторні експерименти *in vitro/in vivo*) є надто повільними, витратними та не можуть охопити увесь спектр потенційних комбінацій. Це обумовлює нагальну необхідність у розробці високопродуктивних, точних та надійних інтелектуальних систем прогнозування DDI, що можуть бути інтегровані в системи підтримки прийняття клінічних рішень.

Останні наукові розробки у цій сфері зосереджені на використанні методів глибинного навчання (Deep Learning, DL). Ці методи продемонстрували здатність виявляти складні, нелінійні патерни взаємодії, що базуються на структурних, молекулярних та фармакологічних характеристиках препаратів. Проте, значна частина існуючих моделей фокусується лише на одному типі даних (наприклад, лише на графовому представленні молекули або лише на її послідовності SMILES-нотації), що обмежує повноту аналізу. Невирішеною проблемою залишається створення мультимодальної архітектури, яка б ефективно інтегрувала та обробляла різнорідні, взаємодоповнюючі джерела інформації про ліки, що є ключовим для підвищення точності та надійності прогнозу DDI.

Актуальність роботи обумовлена тим, що традиційні методи прогнозування DDI - засновані на простих метриках подібності, біологічних експериментах in vitro/in vivo або класичних алгоритмах машинного навчання – часто виявляються недостатніми. Вони не здатні ефективно інтегрувати та інтерпретувати гетерогенні мультимодальні дані (хімічна структура, білкові мішені, метаболічні ферменти, біологічні шляхи) у єдиному контексті. Виникає потреба в архітектурі, яка може динамічно зважувати внесок кожної модальності та виявляти складні, нелінійні міжмодальні залежності.

Метою роботи є розробка та імплементація моделі DDI-TransMDL (Transformer-based Drug-Drug Interaction Prediction Model), яка використовує архітектуру Трансформер-Енкодера для глибокої інтеграції мультимодальних ознак, а також багатоміткове прогнозування 65 унікальних подій взаємодії з високою точністю.

Для досягнення поставленої мети необхідно вирішити такі завдання.

1. Провести огляд та аналіз сучасних методів прогнозування DDI, обґрунтувати вибір Трансформер-архітектури.
2. Сформулювати вимоги до моделі та спроєктувати механізми збору та обробки мультимодальних ознак із бази даних DrugBank, включаючи інтеграцію коефіцієнта Жаккарда.
3. Спроектувати та реалізувати архітектуру Трансформер-Енкодера DDI-TransMDL, включаючи модуль Multi-Head Self-Attention та багатоміткову класифікаційну голову.
4. Розробити та застосувати протокол експериментального дослідження на основі K-Fold Cross-Validation.
5. Провести аналіз процесу навчання, оцінити стійкість моделі та проаналізувати фінальні результати за ключовими метриками (ROC-AUC, AUPR, F1-Score) у Micro та Macro режимах.

Об'єктом дослідження є процес багатоміткового прогнозування DDI.

Предметом дослідження є архітектура та ефективність мультимодальної Трансформер-моделі DDI-TransMDL.

Наукова новизна роботи полягає у першому застосуванні архітектури Трансформер-Енкодера для зваженої та контекстно-залежної інтеграції чотирьох ключових фармакологічних та хімічних модальностей, а також у впровадженні методу інтеграції метрики подібності Жаккарда як експліцитної ознаки взаємодії.

РОЗДІЛ 1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ ТА МЕТОДІВ ПРОГНОЗУВАННЯ ВЗАЄМОДІЙ ЛІКАРСЬКИХ ЗАСОБІВ

1.1 Взаємодії лікарських препаратів: основні поняття та класифікація

1.1.1 Типи лікарських взаємодій

Взаємодія лікарських засобів виникає тоді, коли два або більше препаратів впливають на дію один одного під час одночасного прийому, що може змінювати їхню терапевтичну ефективність або спричиняти небажані наслідки. Розуміння цих взаємодій є фундаментальним для лікарів, адже воно дозволяє безпечно та ефективно керувати схемами лікування. Взаємодії між препаратами загалом поділяються на кілька категорій залежно від їхніх механізмів та наслідків [1-3].

Найчастіше розглядають фармакокінетичні взаємодії. Цей тип взаємодій виникає тоді, коли один препарат впливає на ADME-процеси (від англ. Absorption, Distribution, Metabolism, Excretion - всмоктування, розподіл, метаболізм і виведення). На етапі всмоктування деякі препарати можуть змінювати кислотність шлунка, рухливість кишечника або зв’язування з іншими речовинами. Наприклад, антациди, що містять алюміній або магній, знижують абсорбцію тетрациклінових антибіотиків. Під час розподілу можливі зміни через конкуренцію за білки плазми крові (альбумін, α1-кислотний глікопротеїн). Це може призводити до підвищення вільної (активної) фракції препарату, як у випадку між варфарином і нестероїдними протизапальними засобами. Метаболічні взаємодії - найвідоміші. Багато ліків метаболізуються системою цитохрому P450 [4]. Інгібітори цього ферменту (наприклад, кетоконазол, еритроміцин, грейпфрутовий сік) можуть підвищувати рівень інших препаратів, що метаболізуються цими ферментами, збільшуючи ризик токсичності. Навпаки, індуктори ферментів (рифампіцин, карбамазепін, фенобарбітал) знижують концентрацію

препаратів, що може призвести до втрати терапевтичного ефекту. На етапі виведення взаємодії можуть виникати через зміну ниркової фільтрації або канальцевої секреції. Наприклад, пробенецид зменшує виведення пеніциліну, подовжуючи його дію.

Тісно пов’язаними є фармакодинамічні взаємодії, які виникають тоді, коли препарати впливають на ефекти один одного у місцях дії, не змінюючи їхню концентрацію. Вони можуть бути:

- адитивними (additive), коли ефект двох ліків підсумовується (наприклад, одночасне застосування аспірину та варфарину підвищує ризик кровотеч);
- синергічними (synergistic), коли комбінація дає ефект більший, ніж очікувано від простої суми дій (наприклад, поєднання β-лактамних антибіотиків з аміноглікозидами при тяжких інфекціях)[5];
- антагоністичними (antagonistic), коли один препарат послаблює дію іншого (наприклад, нестероїдні протизапальні засоби можуть зменшувати антигіпертензивний ефект діуретиків або інгібіторів АПФ)[6].

Окрім класичних типів, існують також хімічні взаємодії. Такі взаємодії відбуваються до або після введення препаратів і пов’язані з безпосередньою фізико-хімічною реакцією між речовинами. Вони можуть призвести до інактивування діючої речовини, утворення осаду або зміни розчинності. Наприклад, змішування гепарину з певними антибіотиками в одному шприці може призвести до утворення осаду та втрати активності препаратів. У внутрішньоорганізмових умовах подібні реакції можуть виникати у шлунково-кишковому тракті – наприклад, між препаратами заліза та антацидами, що зменшує біодоступність першого[7].

Ще один важливий аспект – це взаємодії на рівні біологічних мішеней і шляхів. Препарати можуть конкурувати за один і той самий білок-мішень або порушувати споріднені сигнальні шляхи, що призводить до складних

клінічних наслідків. Такі механізми дедалі краще розкриваються завдяки розвитку молекулярної біології та біоінформатики.

Нарешті, взаємодії, що пов’язані з ферментами, виходять за межі лише метаболічних процесів - вони охоплюють випадки, коли вплив ліків на ферментну активність змінює фізіологічні або патологічні процеси. Наприклад, препарати, що модифікують активність ферментів у метаболічних шляхах, можуть впливати як на перебіг захворювань, так і на дію інших ліків.

Складність взаємодій між лікарськими засобами підкреслює важливість інтеграції різноманітних джерел даних - хімічних, біологічних, геномних і клінічних - для ефективного прогнозування та управління лікарськими взаємодіями (DDI). Такі ресурси, як DrugBank [8], FDA Drug Interaction Database [8], PubChem [10], ClinicalTrials.gov [11] та широкі наукові літературні джерела, залишаються ключовими для розуміння цих процесів і розробки безпечніших терапевтичних стратегій.

1.1.2 Клінічне значення прогнозування DDI

Клінічне значення прогнозування взаємодій лікарських засобів (DD [1]) полягає в забезпеченні безпеки пацієнтів та ефективності фармакотерапії. У сучасній медичній практиці поліфармація - тобто одночасне застосування кількох препаратів - є поширеним явищем, особливо серед літніх пацієнтів, а також осіб із хронічними захворюваннями (серцево-судинними, ендокринними, онкологічними). У таких випадках ризик несприятливих взаємодій значно зростає, що може призвести до зниження ефективності лікування, непередбачуваних побічних реакцій або навіть токсичних ускладнень, які нерідко стають причиною госпіталізації або летальних випадків.

Точне прогнозування можливих взаємодій до їхнього клінічного прояву є ключовим елементом профілактики небажаних лікарських реакцій (Adverse Drug Reactions, ADRs). За оцінками Всесвітньої організації охорони здоров'я (ВООЗ), близько 10–20% госпіталізацій у розвинених країнах пов'язані саме з медикаментозними ускладненнями, значна частка яких обумовлена ВЛЗ. Раннє виявлення потенційно небезпечних комбінацій дає змогу оптимізувати терапевтичні стратегії, попередити фатальні наслідки та підвищити якість життя пацієнтів. Наприклад, своєчасне врахування взаємодії між варфарином і антибіотиками макролідного ряду може запобігти внутрішнім кровотечам, а уникнення поєднання інгібіторів МАО із серотонінергічними антидепресантами - серотоніновому синдрому [3].

Сучасні моделі прогнозування DDI сприяють розвитку персоналізованої медицини, оскільки дозволяють адаптувати схеми лікування до індивідуальних особливостей пацієнта.

Такі моделі враховують:

- поточні призначення та можливі фармакокінетичні взаємодії;
- генетичні поліморфізми ферментів, зокрема системи цитохрому P450 (CYP2D6, CYP3A4, CYP2C9) [4], які впливають на швидкість метаболізму лікарських засобів;
- стан печінки та нирок, що визначає кліренс препаратів;
- наявність супутніх захворювань і фізіологічних факторів (вік, стать, маса тіла, дієта).

Врахування цих параметрів дозволяє лікарям індивідуалізувати призначення, зменшити кількість помилок при виборі комбінацій препаратів і знизити ризик побічних ефектів CDSS, що використовують такі моделі, уже інтегровані у медичні інформаційні системи США, ЄС та Японії. Вони автоматично попереджають лікаря про потенційно небезпечні комбінації ще під час електронного призначення ліків [12].

Застосування машинного навчання (ML) та глибинного навчання (DL) докорінно змінює підходи до прогнозування DDI. Алгоритми цих систем можуть:

- аналізувати великі масиви даних із хімічними структурами, фармакологічними властивостями та результатами клінічних досліджень;
- виявляти приховані взаємозв'язки між ліками, мішенями, метаболітами та побічними ефектами;
- прогнозувати нові або рідкісні взаємодії, не описані в літературі.

Такі системи базуються на даних з ресурсів DrugBank, PubChem, FAERS (FDA Adverse Event Reporting System) та ClinicalTrials.gov [16–18] . Приклади моделей включають DeepDDI, DeepDrug, BioBERT-DDI [13] , які демонструють високу точність у виявленні потенційно небезпечних комбінацій.

1.1.3 Фармакокінетичні та фармакодинамічні аспект

Взаємодії між лікарськими засобами становлять одну з ключових проблем сучасної фармакології та клінічної медицини. Вони можуть виникати через два основних біологічних механізми – фармакокінетичний і фармакодинамічний. Ці два механізми описують різні, але взаємопов'язані способи, за допомогою яких ліки впливають на дію одне одного в організмі людини. Розуміння цих процесів є критично важливим для запобігання небажаним побічним ефектам, оптимізації комбінованої терапії та підвищення безпеки лікування.

Фармакокінетичні взаємодії виникають тоді, коли один препарат змінює процеси всмоктування, розподілу, метаболізму або виведення іншого, тобто впливає на його фармакокінетичний профіль. У результаті цього

змінюється концентрація діючої речовини в крові та тканинах, а отже - і сила, і тривалість її дії. Такі взаємодії можуть проявлятися у різний спосіб. Один препарат здатний пригнічувати або активувати ферменти печінкового метаболізму, зокрема ізоформи цитохрому Р450 (CYP3A4, CYP2D6, CYP2C9, CYP1A2). Наприклад, інгібітори цих ферментів, як-от кларитроміцин або метронідазол, можуть збільшувати концентрацію варфарину в плазмі, що підвищує ризик кровотеч. У протилежних випадках індуктори ферментів, такі як рифампіцин або карбамазепін, прискорюють метаболізм інших препаратів, зменшуючи їхню терапевтичну дію.

Інший механізм фармакокінетичної взаємодії може бути пов’язаний із впливом на транспортні білки, наприклад, на Р-глікопротеїн, який регулює проникнення лікарських речовин у клітини та їх виведення з організму. Також взаємодії можуть виникати через зміни кислотності шлункового соку, що впливають на абсорбцію деяких препаратів (наприклад, антациди знижують засвоєння кетоконазолу), або через зміну ступеня зв’язування з білками плазми, що змінює розподіл активної фракції ліків у тканинах. Ці механізми мають особливе значення у випадках поліфармації, коли пацієнт приймає декілька лікарських засобів одночасно. Найчастіше така ситуація спостерігається у літніх людей або пацієнтів із хронічними захворюваннями, де навіть незначна зміна фармакокінетичних параметрів може спричинити серйозні клінічні наслідки.

Фармакодинамічні взаємодії мають іншу природу. Вони відбуваються не через зміну концентрації ліків у крові, а через їх вплив на одні й ті самі або взаємопов’язані фізіологічні системи, рецептори чи сигнальні шляхи. Інакше кажучи, це взаємодії, що стосуються того, “що робить препарат з організмом”, а не того, “що організм робить із препаратом”. Такі взаємодії можуть бути адитивними, коли дія препаратів підсумовується; синергічними, коли один препарат посилює ефект іншого; або антагоністичними, коли один знижує дію іншого.

Прикладом адитивного ефекту є комбінація кількох антигіпертензивних засобів, наприклад бета-блокатора та інгібітора ангіотензинперетворювального ферменту, що може призвести до надмірного зниження артеріального тиску. Синергічні взаємодії спостерігаються у комбінованій антибактеріальній терапії, коли поєднання амоксициліну з клавулановою кислотою забезпечує підвищену ефективність лікування завдяки блокуванню бактеріальних ферментів, що руйнують антибіотик. У свою чергу, антагоністичний ефект проявляється, наприклад, при одночасному застосуванні β-агоністів і β-блокаторів, які діють на один і той самий рецептор, але з протилежним результатом.

Окрему увагу заслуговують випадки, коли фармакодинамічні взаємодії стають клінічно небезпечними, навіть якщо фармакокінетичні зміни відсутні. Поєднання нестероїдних протизапальних засобів із антикоагулянтами може значно підвищити ризик шлунково-кишкових кровотеч, оскільки обидва класи препаратів впливають на різні механізми гемостазу, але кінцевий ефект – ослаблення зсідання крові – є спільним. Водночас, коли такі взаємодії ретельно контролюються, вони можуть бути терапевтично корисними. Прикладом є комбінована хіміотерапія при онкологічних або вірусних захворюваннях, де поєднання препаратів із різними механізмами дії дозволяє досягти більшого лікувального ефекту з меншою токсичністю.

Розуміння обох типів взаємодій – фармакокінетичних і фармакодинамічних - є основою сучасної раціональної фармакотерапії. Це знання дає змогу лікарям передбачати потенційні ризики ще до того, як вони проявляться клінічно, і підбирати безпечніші та ефективніші комбінації препаратів. Застосування терапевтичного моніторингу лікарських засобів, корекція дозування та вибір альтернативних шляхів лікування допомагають уникати серйозних ускладнень і покращують результати терапії.

Сьогодні до цієї проблеми активно залучаються сучасні обчислювальні технології. Моделі машинного та глибинного навчання дозволяють аналізувати великі обсяги фармакологічних, хімічних і клінічних даних, щоб

виявляти потенційно небезпечні взаємодії ще до того, як препарат потрапить у клінічну практику. Інтеграція таких підходів із базами даних, як-от DrugBank, PubChem, FAERS або ClinicalTrials.gov, робить прогнозування більш точним і швидким. Це особливо важливо у світлі розвитку персоналізованої медицини, де вибір комбінації ліків має враховувати не лише фармакологічні властивості препаратів, а й індивідуальні особливості пацієнта – генетичні, метаболічні, фізіологічні.

Отже, фармакокінетичні та фармакодинамічні взаємодії є двома фундаментальними сторонами складного процесу взаємодії лікарських засобів. Їхнє глибоке розуміння дозволяє лікарям і науковцям не лише передбачати небажані ефекти, а й свідомо використовувати потенційно вигідні комбінації для підвищення ефективності лікування. Знання цих механізмів, доповнене сучасними біоінформатичними інструментами, формує основу безпечної та науково обґрунтованої фармакотерапії майбутнього.

1.2 Огляд існуючих підходів до прогнозування DDI

1.2.1 Традиційні методи на основі баз знань

До появи методів машинного та глибинного навчання саме знання-орієнтовані підходи становили найраніші та найпоширеніші обчислювальні стратегії для прогнозування взаємодій між лікарськими засобами. Ці методи ґрунтувалися на експертно сформованих біомедичних даних, фармакологічних правилах і механістичних знаннях про те, як лікарські речовини взаємодіють у межах біологічних систем[14]. На відміну від сучасних моделей, які виявляють закономірності на основі великих масивів даних, знання-орієнтовані системи використовували явну логіку, побудовану

на вже відомих зв’язках, зафіксованих у базах даних, науковій літературі та біохімічних онтологіях.

Перші системи прогнозування DDI [15] зазвичай створювалися як правилово-орієнтовані експертні системи або семантичні фреймворки логічного виведення. Їхня робота спиралася на бази даних, такі як DrugBank, KEGG чи PubChem, що містять структуровану інформацію про хімічні властивості лікарських речовин, їхні молекулярні мішені, ферменти, транспортні білки та метаболічні шляхи. Алгоритм таких систем полягав у виявленні збігів або перетинів у цих характеристиках. Наприклад, якщо два препарати метаболізуються одним і тим самим ферментом, як-от CYP3A4, система могла зробити висновок про потенційне конкурентне зв’язування та зміну швидкості метаболізму, що може призвести до токсичності або зниження ефективності лікування. Такі знання кодувалися у вигляді правил, які використовувалися для формування попереджень про можливі лікарські взаємодії.

Подібні системи стали основою перших CDSS, які інтегрували фармаконаглядові дані та автоматично видавали попередження про небезпечні комбінації під час призначення ліків. Цей підхід дозволяв лікарям швидко отримувати інтерпретовану інформацію про потенційні ризики, зменшуючи ймовірність помилок при поліфармації.

Одним із ключових компонентів таких методів було використання знаннєвих графів (knowledge graphs). У цих графах лікарські засоби, білки, біохімічні шляхи, побічні ефекти та інші сутності відображалися у вигляді вузлів і зв’язків між ними. Завдяки такому мережевому представленню система могла робити висновки навіть тоді, коли прямі експериментальні докази були відсутні. Наприклад, якщо препарат A взаємодіє з препаратом B, а препарат B взаємодіє з препаратом C, то за допомогою логічного висновування система могла передбачити можливу опосередковану взаємодію між A і C. Цей принцип транзитивного висновування був широко використаний у ранніх графових моделях DDI-прогнозування[16].

Попри прозорість і високу інтерпретованість таких підходів, вони мали й суттєві обмеження. Точність прогнозів безпосередньо залежала від повноти та якості доступних баз даних, а також від актуальності інформації про лікарські засоби. При появі нових сполук або невідомих комбінацій ефективність моделей різко знижувалася. Крім того, створення та підтримка таких систем вимагала значних людських ресурсів, оскільки експерти мали постійно оновлювати правила, перевіряти джерела та забезпечувати відповідність новим науковим даним. Через це знання-орієнтовані підходи поступово поступилися місцем більш гнучким даних-орієнтованим (data-driven) моделям, зокрема алгоритмам машинного та глибинного навчання, які здатні автоматично виявляти закономірності без попереднього формулювання правил.

Проте, незважаючи на свої недоліки, знання-орієнтовані методи не втратили значення. Вони й надалі залишаються важливою складовою сучасної біомедичної інформатики, оскільки забезпечують механістичне пояснення процесів, які складно інтерпретувати за допомогою виключно статистичних моделей. У сучасних гібридних системах DDI-прогнозування знання з таких баз, як DrugBank, PharmGKB чи ChEMBL, інтегруються як онтологічні каркаси або структуровані пріори, що підвищують інтерпретованість, надійність і прозорість результатів машинного аналізу.

Таким чином, знання-орієнтовані підходи заклали фундамент для розвитку сучасних інтелектуальних систем прогнозування лікарських взаємодій. Вони не лише сприяли формуванню перших електронних інструментів підтримки клінічних рішень, але й стали основою для нової хвилі досліджень, у яких символічне міркування поєднується з алгоритмами штучного інтелекту, формуючи основу для пояснюваних та клінічно значущих моделей у цифровій фармакології.

1.2.2 Методи машинного навчання для прогнозування взаємодії

Машинне навчання стало одним із найпотужніших інструментів для прогнозування взаємодій між лікарськими засобами[17], оскільки воно здатне аналізувати великі обсяги біомедичних даних, виявляючи приховані закономірності, які недосяжні для традиційних підходів. Сучасні моделі машинного навчання дозволяють передбачати потенційні фармакокінетичні та фармакодинамічні взаємодії ще на доклінічному етапі розробки препаратів, що значно підвищує безпеку пацієнтів і зменшує кількість побічних ефектів.

Прогнозування DDI за допомогою машинного навчання ґрунтується на аналізі структурних, біохімічних та фармакологічних ознак препаратів. Моделі будуються на основі ознак, що включають хімічні дескриптори, експресію білкових мішеней, біологічні шляхи та метаболічні ферменти. Алгоритми, як-от SVM, RF, Gradient Boosting [17], а також нейронні мережі, навчаються на великих наборах даних, таких як DrugBank, TWOSIDES або BIOSNAP [19], створюючи здатність узагальнювати закономірності між молекулярними властивостями лікарських засобів.

Окремий напрямок становлять глибокі нейронні мережі (Deep Learning), зокрема CNN, GNN [18] та transformer-архітектури, які здатні працювати з мультиомними та текстовими даними. Такі моделі розпізнають складні взаємозв’язки між молекулярними структурами, шляхами метаболізму та біологічними системами. Наприклад, GNN-підходи представляють лікарські препарати як вузли у графі з ребрами, що позначають їхні відомі взаємодії, а трансформери використовують семантичну інформацію з наукових публікацій для відкриття нових DDI.

Клінічна цінність цих моделей полягає у прогнозуванні ризиків комбінованої фармакотерапії, особливо у пацієнтів з поліпрепарацією. Системи, засновані на машинному навчанні, інтегруються в клінічні рішення

(CDSS), дозволяючи лікарям своєчасно попереджати небезпечні комбінації ліків і вибирати оптимальні терапевтичні схеми. Крім того, у фармацевтичних дослідженнях такі системи використовуються для in silico оцінки нових препаратів, що скорочує час і вартість клінічних випробувань.

Завдяки швидкому розвитку машинного навчання прогнозування взаємодій між препаратами переходить від емпіричних методів до автоматизованих систем, які поєднують фармакологічні знання, біомолекулярні дані та алгоритмічне моделювання. Таким чином, ШІ відкриває нову еру у створенні безпечних і персоналізованих лікарських терапій, що є ключовим напрямом сучасної медичної інформатики.

1.3 Мультимодальні моделі в біомедичній інформатиці

1.3.1 Концепція мультимодального навчання

Мультимодальне навчання (multimodal learning) - це підхід, що передбачає інтеграцію та одночасну обробку кількох типів даних або джерел інформації з метою побудови більш точних, надійних і комплексних моделей прогнозування. У контексті DDI цей підхід полягає у поєднанні різнорідних характеристик лікарських препаратів - таких як хімічна структура, молекулярні мішені, біохімічні шляхи, ферментативна активність, профілі побічних ефектів та клінічні анотації.

Основна ідея мультимодального навчання полягає у створенні моделей, здатних витягувати взаємодоповнювальну інформацію з різнорідних типів даних. Це дозволяє виявляти приховані закономірності та сигнали взаємодій, які залишаються непоміченими у випадку використання лише однієї модальності (унімодального підходу). Наприклад, SMILES-послідовності відображають молекулярну будову речовини, тоді як інформація про білкові мішені, метаболічні шляхи чи ферменти відображає її функціональний і

біологічний контекст. Об’єднання цих рівнів дозволяє формувати більш повне уявлення про потенційні механізми взаємодій між препаратами.

Для реалізації такого підходу використовують спеціалізовані нейронні архітектури [21], здатні працювати з різними типами даних - від хімічних графів до послідовностей або текстових описів. Серед найпоширеніших моделей – MCNN [20], GNN з механізмами уваги (attention mechanisms), а також автоенкодери та трансформерні архітектури, що навчаються спільним просторам представлення (joint embeddings). Завдяки цьому модель здатна інтегрувати молекулярні, фармакологічні та біологічні ознаки у єдиний репрезентативний простір, де схожі за механізмом дії лікарські засоби розташовуються поруч, навіть якщо вони структурно відрізняються.

У практичній реалізації мультимодальні DDI-моделі зазвичай складаються з окремих підмереж (subnetworks) для кожного типу даних. Кожна підмережа відповідає за попередню обробку та кодування певних ознак - наприклад, одна мережа аналізує хімічну структуру, інша працює з білковими або метаболічними шляхами, а третя - з фармакогеномними чи клінічними профілями. Потім отримані представлення об’єднуються за допомогою шарів злиття (fusion layers) – таких як конкатенація, вагова або увагова інтеграція (attention-based fusion). У результаті формується єдине інтегроване представлення препарату або пари препаратів, яке використовується для прогнозування ймовірності їхньої взаємодії.

Такий підхід має кілька ключових переваг. По-перше, він дозволяє використовувати сильні сторони кожного типу даних, компенсуючи обмеження інших. По-друге, мультимодальні моделі краще узагальнюють нові дані, що підвищує стійкість (robustness) і здатність до перенесення (transferability) між наборами даних. По-третє, мультимодальне навчання сприяє зменшенню переобладнання (overfitting), оскільки модель не спирається на один домінуючий тип ознак. Результати нещодавніх досліджень показали, що такі моделі суттєво перевершують унімодальні підходи за точністю прогнозування потенційно небезпечних DDI, особливо

при використанні комбінованих структурних, біологічних і фармакологічних даних.

Отже, мультимодальне навчання в фармакоінформатиці є просунутим напрямом, який відображає природну складність біомедичних процесів і дозволяє досягати вищої прогностичної потужності та пояснюваності результатів. Його впровадження сприяє розвитку персоналізованої медицини, покращує оцінку безпеки лікарських засобів та відкриває нові можливості для дослідження молекулярних механізмів фармакологічних взаємодій.

1.3.2 Необхідні ознаки для розв'язку поставленої задачі

Комплексний аналіз лікарських засобів і їхніх взаємодій базується на використанні кількох категорій даних, що в сукупності описують хімічні властивості, біологічні функції та клінічні наслідки застосування препаратів [24]. Ці різнорідні типи інформації формують основу для побудови прогностичних моделей у галузі фармакоінформатики та машинного навчання, спрямованих на виявлення потенційних DDI.

Найбазовішим типом інформації є дані про хімічну структуру, що відображають молекулярний склад, типи зв'язків і просторову організацію лікарських речовин. Такі дані зазвичай представлені у вигляді SMILES, InChI-кодів або молекулярних графів, які моделюють атоми як вузли, а зв'язки між ними - як ребра. Відкриті бази даних, такі як DrugBank, ChEMBL [22] і PubChem, забезпечують широкий доступ до структурної інформації про лікарські засоби, дозволяючи генерувати молекулярні дескриптори, фінгерпринти, підструктури та фізико-хімічні властивості (масу, полярність, гідрофобність тощо). Для препарату Аторвастатин, SMILE-рядок може виглядати так:

CC(C)c1ccc(C(O)Cc2ccccc2)c1NC(=O)c1c(C)c(C)c(c3ccccc3)n1CC(=O)O.

Завдяки таким дескрипторам моделі машинного навчання можуть передбачати можливість взаємодій на основі структурної схожості та потенційної спорідненості до спільних мішеней. Наприклад, препарати з подібними ароматичними групами або зарядженими фрагментами можуть конкурувати за зв’язування з тим самим ферментом, що часто є основою фармакокінетичних DDI.

Не менш важливими є біологічні та біохімічні дані, які встановлюють зв’язки між лікарськими засобами та їхніми молекулярними мішенями - білками, ферментами або рецепторами. Інформація про взаємодії “препарат–мішень” (drug–target interactions, DTI) та “препарат–фермент” отримується з таких ресурсів, як BindingDB, UniProt, PharmGKB і KEGG.

Дані про мішені описують первинну біологічну дію препарату, вказуючи набір білків-мішеней (наприклад, рецепторів, кіназ, іонних каналів), з якими він безпосередньо зв'язується або на які впливає. Призначенням target є визначення основного механізму дії ліків. Спільна дія на мішені може призвести до підсилення (синергії) або небажаних побічних ефектів. До прикладу, препарат, який пливає на Adrenoceptor-beta-1 матиме target такого вигляду ID P08588.

Особливу роль відіграє інформація про сімейство ферментів цитохрому P450, оскільки воно визначає метаболічну долю більшості лікарських речовин. Наприклад, інгібітори CYP3A4 можуть підвищувати концентрацію субстратів цього ферменту, що призводить до токсичних ефектів - типовий приклад метаболічної взаємодії.

Ще одна важлива категорія - це дані про сигнальні та метаболічні шляхи, де препарати відображаються в контексті біологічних мереж. Такі моделі показують, як кілька лікарських засобів можуть впливати на спільні молекулярні каскади, ферментативні ланцюги або спільні вузли регуляції. У цьому підході дані структуруються у вигляді графів або мереж, де вузлами

виступають препарати, білки, гени чи побічні ефекти, а зв’язки між ними описують відомі або передбачувані взаємодії. До прикладу, препарат може впливати на шлях апоптозу (hsa04210) та шлях метаболізму ліпідів (hsa00561).

Мережеві моделі дозволяють робити структурні висновки про можливі механізми DDI навіть у випадках, коли прямі експериментальні дані відсутні. Наприклад, якщо два препарати впливають на різні білки в межах однієї сигнальної мережі, їх одночасне застосування може викликати синергічний або антагоністичний ефект.

Текстові та семантичні джерела доповнюють структуровані бази, забезпечуючи контекстуальну інформацію з наукової літератури, клінічних звітів, патентів і регуляторних документів. Використовуючи методи NLP і сучасні трансформерні моделі (наприклад, BioBERT [25], PubMedBERT або SciBERT [26]), можна автоматично вилучати відомості про нові чи недокументовані взаємодії, механістичні гіпотези або оновлення фармакологічних класифікацій.

Такі підходи допомагають зменшити інформаційний розрив між експериментальними та клінічними даними, забезпечуючи безперервне оновлення знань про DDI.

У сучасній фармакоінформатиці всі перелічені типи даних інтегруються за допомогою мультимодальних навчальних архітектур. Поєднання хімічних, біологічних, мережевих, клінічних і текстових джерел дає змогу створювати багатовимірні представлення лікарських засобів, що враховують їхню хімію, біологію та поведінку в організмі. Такий підхід не лише підвищує точність прогнозування взаємодій, а й покращує інтерпретованість моделей, забезпечуючи більш глибоке розуміння фармакологічних процесів на молекулярному та системному рівнях.

1.3.3 Архітектури мультимодальних нейронних мереж

Мультимодальні нейронні мережі інтегрують різнорідні джерела даних у межах єдиної глибинної архітектури, що дозволяє моделі вловлювати доповнювальні взаємозв'язки між різними типами ознак - хімічними, біологічними та структурними характеристиками лікарських засобів. У контексті прогнозування взаємодій між лікарськими засобами (drug–drug interactions, DDI) такі архітектури розробляються для паралельної обробки різних модальностей даних - хімічних представлень (SMILES), білкових мішеней, ферментів і біохімічних шляхів. Кожна модальність проходить через окремий навчальний модуль, після чого результати об'єднуються на рівні мультимодального злиття (fusion layer) для формування спільного представлення, що використовується для точної класифікації або передбачення типу взаємодії.

Однією з найбільш відомих мультимодальних моделей є DDIMDL [27] (Deep Drug Interaction Multimodal Deep Learning). Вона складається з чотирьох підмереж, кожна з яких відповідає за певну модальність - хімічні підструктури, мішені, ферменти та метаболічні шляхи. Кожна підмережа реалізована у вигляді щільних або згорткових шарів (dense/convolutional layers) і навчається формувати високорозмірні векторні представлення (embeddings) відповідного типу даних.

На етапі злиття отримані представлення об'єднуються у спільний латентний простір, який використовується для багатокласового прогнозування DDI-подій. Такий підхід дозволяє моделі виявляти взаємозв'язки між структурними, біологічними та функціональними ознаками, що забезпечує вищу точність та інтерпретованість порівняно з одномодальними системами.

Інша ефективна архітектура - MCNN-DDI[20] (Multimodal Convolutional Neural Network) – реалізує кілька паралельних CNN-шляхів, кожен із яких

відповідає за окремий тип ознак (SMILES, мішені, ферменти, шляхи). Після самостійного навчання кожного шляху їхні вихідні вектори об’єднуються за допомогою щільних шарів злиття (dense fusion layers).

Таке поєднання дозволяє спочатку навчати локальні ознаки всередині модальності, а потім виявляти кореляції між різними типами даних на етапі злиття. За результатами тестування на наборах даних із DrugBank, ця архітектура досягла точності 90,00% і AUPR 94,78%, що перевищує показники одномодальних базових моделей.

Мультимодальні нейронні архітектури демонструють свою ефективність завдяки поєднанню спеціалізованих енкодерів для кожної модальності з етапом злиття (fusion stage), що може бути реалізований через конкатенацію, механізми уваги або графове агрегування.

Використовуючи дані типів SMILES, мішеней (targets), ферментів і біохімічних шляхів, моделі на кшталт DDIMDL та MCNN-DDI створюють базу для сучасних трансформерних або графових моделей, здатних виявляти структурно-біохімічні закономірності між модальностями. У численних експериментах ці підходи стабільно перевершують одномодальні методи, оскільки враховують багатогранну природу лікарських взаємодій на молекулярному та системному рівнях.

Висновки до розділу 1

У першому розділі було проведено комплексний аналіз предметної області прогнозування взаємодій між лікарськими засобами (DDI) та розглянуто еволюцію наукових і технологічних підходів у цій сфері. Визначено основні типи лікарських взаємодій - фармакокінетичні, фармакодинамічні, хімічні та біологічні - а також показано їхнє клінічне

значення для забезпечення безпеки та ефективності фармакотерапії, особливо в умовах поліфармації.

Проаналізовано традиційні методи прогнозування DDI, що базуються на знаннях та експертних системах, які використовували логічні правила, онтології та бази даних (DrugBank, KEGG, PubChem). Хоча ці системи мали високу інтерпретованість, їхня ефективність обмежувалася неповнотою даних і необхідністю постійного оновлення знань.

Окрему увагу приділено мультимодальним моделям, які поєднують різнорідні джерела даних - хімічні, біологічні, клінічні та текстові - у межах єдиної аналітичної архітектури. Такі підходи забезпечують глибше розуміння механізмів взаємодії препаратів і відкривають нові можливості для персоналізованої медицини.

Отже, у розділі доведено, що інтеграція мультимодальних даних із сучасними методами штучного інтелекту формує нову парадигму прогнозування лікарських взаємодій. Це сприяє підвищенню безпеки лікування, оптимізації терапевтичних стратегій і розвитку цифрової фармакології як ключового напряму сучасної медичної інформатики.

РОЗДІЛ 2 МЕТОДИ ГЛИБОКОГО НАВЧАННЯ ТА ІНТЕГРАЦІЇ МУЛЬТИМОДАЛЬНИХ ДАНИХ У МОЛЕКУЛЯРНИХ ЗАДАЧАХ

2.1 Математичні основи глибокого навчання

2.1.1 Нейронні мережі та функції активатори

Нейронні мережі становлять математичну основу глибинного навчання. По суті, це параметричні апроксиматори функцій, здатні навчатися складних нелінійних відображень між вхідними даними та вихідними прогнозами. Базова feedforward neural network або MLP [28], може бути описана як композиція афінних перетворень та нелінійних активаційних функцій.

Для вхідного вектора одношарова мережа обчислює:

, (2.1)

де – матриці ваг, – вектори зсуву (bias), а – нелінійна активаційна функція, застосована до кожного елемента.

Параметри і оптимізуються за допомогою градієнтних методів, найпоширенішим з яких є Stochastic Gradient Descent [29].

З математичної точки зору, нейронні мережі функціонують як універсальні апроксиматори. Згідно з теоремою універсальної апроксимації, навіть одношарова нейронна мережа з кінцевою кількістю нейронів може апроксимувати будь-яку неперервну функцію на компактній підмножині , якщо правильно обрано ваги та активаційні функції. Ця властивість пояснює здатність нейронних мереж вивчати складні закономірності в різних типах даних – хімічних, біологічних, текстових чи клінічних.

Виразність (expressiveness) моделі залежить від її глибини (кількості шарів) та ширини (кількості нейронів у шарі), що впливають як на здатність до апроксимації, так і на обчислювальну складність.

Активаційні функції надають мережі нелінійності, яка дозволяє захоплювати складні взаємозв'язки у даних. Без них, тобто у разі використання лише лінійних перетворень, уся мережа математично зводилася б до одного лінійного оператора.

До найпоширеніших активацій належать [29]:

, (2.2)

, (2.3)

(2.4)

Формула (2.2) – сигмоїда, стискає значення у діапазон (0,1). Історично важлива, проте страждає від проблеми згасання градієнтів у глибоких мережах. Формула (2.3) – гіперболічний тангенс, який центрує активації навколо нуля, зменшуючи насичення, властиве сигмоїді. Формула (2.4) – ReLU, стандарт у сучасних архітектурах через простоту, ефективність та розріджені активації, що прискорюють навчання.

Для шару роботу мережі можна записати як:

, (2.5)

де – вихід попереднього шару (або вхідний вектор для), а – вхід наступного шару. Зворотне поширення помилки (backpropagation) обчислює градієнти втрат за всіма параметрами, застосовуючи правило ланцюга, що дозволяє ефективно тренувати глибокі мережі з великою кількістю параметрів.

З математичного погляду, навчання нейронної мережі можна розглядати як мінімізацію неконвексної функції втрат у багатовимірному просторі параметрів.

Такі функції часто мають множинні локальні мінімуми та сідлові точки, однак емпіричні дослідження показують, що правильно регуляризовані моделі, навчені стохастичними методами, зазвичай збігаються до розв’язків, які добре узагальнюються (generalize well). Методи batch normalization та dropout додатково стабілізують навчання, запобігаючи надмірному пристосуванню до тренувальних даних.

Таким чином, нейронні мережі є фундаментальною математичною архітектурою глибинного навчання. Поєднуючи лінійні перетворення, нелінійні активації та градієнтну оптимізацію, вони виступають універсальними моделями, здатними апроксимувати складні залежності у різних типах даних.

2.1.2 Методи оптимізації та регуляризації

Оптимізація та регуляризація становлять дві фундаментальні основи сучасних систем глибокого навчання. Оптимізація забезпечує ефективне навчання нейронних мереж шляхом мінімізації функції втрат відносно параметрів моделі , тоді як регуляризація гарантує здатність моделі узагальнювати результати поза межами тренувальних даних, запобігаючи перенавчанню та підвищуючи стійкість. Разом ці процеси дають змогу досягати високої точності прогнозування при збереженні стабільності та інтерпретованості навіть у складних задачах.

Навчання глибоких нейронних мереж полягає у пошуку мінімумів сильно не опуклої поверхні втрат, визначеної над мільйонами або навіть мільярдами параметрів. Найбільш базовим підходом є градієнтний спуск, у

якому параметри оновлюються ітеративно в напрямку, протилежному градієнту функції втрат:

, (2.6)

де – це швидкість навчання, що визначає розмір кроку оновлення.

На практиці найчастіше застосовують SGD або його міні-батч варіант, які забезпечують кращу обчислювальну ефективність та сприяють узагальненню результатів за рахунок випадкової вибірки даних. Для покращення стабільності та швидкості збіжності використовується метод моменту, що враховує середнє значення попередніх градієнтів:

, (2.7)

, (2.8)

де – коефіцієнт моменту, який визначає вагу попередніх градієнтів [30].

Сучасні алгоритми оптимізації також адаптують швидкість навчання автоматично для кожного параметра. AdaGrad [31] змінює швидкість навчання залежно від накопичених величин градієнтів, що підвищує ефективність для розріджених даних. RMSProp [32] вводить коефіцієнт експоненційного згасання, щоб запобігти надто швидкому зменшенню швидкості навчання. Adam [32] поєднує переваги моменту та адаптивного регулювання швидкості, завдяки чому є одним з найпопулярніших оптимізаторів у глибокому навчанні. Ці методи відіграють ключову роль у біоінформатиці, зокрема у прогнозуванні DDI, де дані мають високу варіабельність і багатомодальність.

Якщо оптимізація визначає як модель навчається, то регуляризація визначає що саме вона засвоює. Регуляризація запобігає перенавчанню -

ситуації, коли модель запам’ятовує шум або випадкові особливості тренувальних даних, а не вивчає закономірності, що узагальнюються.

Базовим методом є L2-регуляризація або згасання ваг (weight decay), яка штрафує великі значення параметрів, додаючи до функції втрат член :

$$_{reg} \quad . \qquad (2.9)$$

Цей підхід сприяє формуванню простіших моделей із кращою узагальнюючою здатністю.

Іншим поширеним методом є dropout [33], який під час навчання випадково “вимикає” нейрони з певною ймовірністю. Це запобігає надмірній залежності між нейронами (коадаптації) та стимулює мережу формувати надлишкові, більш стійкі представлення. Dropout особливо ефективний у щільних і згорткових архітектурах, що використовуються в біомедичних задачах з обмеженими обсягами даних.

Ще один важливий метод – нормалізація батчів (Batch Normalization) [34], яка стандартизує вихідні значення кожного шару. Це стабілізує розподіл активацій у процесі навчання, прискорює збіжність та діє як м’який регуляризатор, знижуючи ризик перенавчання.

До додаткових підходів належать рання зупинка (early stopping) - припинення навчання, коли точність на валідаційній вибірці перестає зростати, - та аугментація даних (data augmentation), яка штучно збільшує обсяг тренувальної вибірки через обертання, масштабування або додавання шуму. Сучасні техніки, такі як згладжування міток (label smoothing) та стохастична глибина (stochastic depth), забезпечують ще кращу узагальненість у випадках невизначених або неповних даних.

2.1.3 Функції втрат та класифікації

Функції втрат є математичним фундаментом процесу навчання нейронних мереж. Вони надають кількісну оцінку того, наскільки передбачення моделі відповідають реальним результатам. Під час навчання глибокої моделі основна мета полягає у мінімізації цього розходження - зазвичай представленого скалярним значенням втрат - шляхом поступового коригування параметрів моделі за допомогою градієнтних методів оптимізації. Чим менше значення функції втрат, тим точніше модель відображає структуру вихідних даних.

Нехай маємо набір даних , де - це вектори ознак (вхідні дані), а - відповідні істинні мітки. Нейронна мережа генерує прогноз , параметризований вагами . Функція втрат визначає загальну ціну (помилку) цих передбачень:

. (2.10)

Мінімізація за допомогою алгоритмів, таких як SGD забезпечує поступове вдосконалення параметрів у напрямку зменшення похибки - процес, відомий як емпірична мінімізація ризику.

Тип функції втрат залежить від задачі – регресії, класифікації чи генеративного моделювання.

Коли цільовими значеннями є неперервні величини (наприклад, прогнозування дози лікарського препарату або фізико-хімічних властивостей молекули), найчастіше використовуються такі функції: (2.11) Середньоквадратична похибка сильно карає великі відхилення, є диференційовною та добре підходить для градієнтного навчання та (2.12)

Середня абсолютна похибка більш стійка до викидів порівняно з MSE і орієнтується на медіанне, а не середнє відхилення.

–, (2.11)

–. (2.12)

У задачах класифікації – наприклад, прогнозуванні типів DDI – застосовуються ймовірнісні функції втрат, такі як Бінарна крос-ентропія (2.13) для двокласових задач, Категоріальна крос-ентропія (2.14) для багатокласових задач.

– , (2.13)

. (2.14)

Категоріальна крос-ентропія використовується разом із softmax-активацією на вихідному шарі, щоб оцінити відмінність між розподілом передбачених і реальних класів. Для задач з дисбалансом класів, що часто трапляється в біомедичних наборах даних, застосовується фокальна функція втрат (Focal Loss) [35], яка зменшує вагу “легких” прикладів і зосереджується на складних, рідкісних зразках.

Вибір відповідної функції втрат і класифікаційної функції є критично важливим для досягнення балансу між зміщенням, варіативністю та стійкістю моделі. У біоінформатиці дизайн функції втрат також впливає на здатність моделі працювати з шумними, незбалансованими або багатомодальними даними.

У складних моделях для передбачення взаємодій лікарських засобів, що об’єднують дані про шляхи метаболізму, ферменти та молекулярні структури, часто застосовують композитні або зважені мультизадачні

функції втрат. Вони дозволяють одночасно оптимізувати кілька пов’язаних підзадач, підвищуючи точність та узагальненість моделі.

2.2 Архітектури для обробки молекулярних даних

2.2.1 Графові нейронні мережі

Графові нейронні мережі (GNN) є одними з найпотужніших архітектур для обробки молекулярних даних, де сполуки, білки та їх взаємодії природним чином описуються у вигляді графів. У такому поданні атоми розглядаються як вузли (nodes), а хімічні зв’язки - як ребра (edges), що дозволяє GNN більш ефективно моделювати реляційні та топологічні властивості молекул, ніж методи, побудовані на регулярних решітках, такі як CNN. Завдяки ітераційному механізму передачі повідомлень (message passing) та агрегації, GNN навчаються створювати інформативні векторні подання (embeddings), які відображають як локальні хімічні середовища, так і глобальний контекст молекули. Це робить їх центральним інструментом у сучасній комп’ютерній хімії та прогнозуванні DDI.

У центрі роботи GNN лежить концепція передачі повідомлень (message passing framework). Кожен вузол у молекулярному графі оновлює своє векторне представлення протягом кількох ітерацій , агрегуючи інформацію від сусідніх вузлів :

, (2.15)

, (2.16)

де – функція генерації повідомлення, а – функція оновлення; обидві мають параметри, що навчаються. Після ітерацій мережа формує

векторне представлення всього графа за допомогою диференційованої функції зчитування (readout), наприклад: R({ | v V}).

Отримане представлення використовується як вхід для подальших завдань – передусім для прогнозування властивостей молекул або класифікації DDI.

Існує кілька важливих варіантів GNN, розроблених спеціально для молекулярного моделювання:

- Graph Convolutional Networks (GCN) - наближають спектральні згортки на графах, що дозволяє поширювати повідомлення між вузлами через нормалізовану матрицю суміжності.
- Graph Attention Networks (GAT) - розширюють GCN, вводячи механізм уваги (attention), який надає різну вагу зв’язкам між вузлами, дозволяючи моделі фокусуватися на хімічно чи функціонально значущих атомах і зв’язках.
- Message Passing Neural Networks (MPNN) - узагальнюють ці підходи, відокремлюючи функції передачі повідомлень та оновлення, що забезпечує високу точність у прогнозуванні токсичності, розчинності та біоактивності молекул.

Попри свою ефективність, GNN стикаються з типовими проблемами over-smoothing (коли подання вузлів стають надто подібними між шарами) та over-squashing (коли далекі залежності стискаються у вузьке векторне представлення), що є особливо актуальним для великих молекулярних графів.

Новітні дослідження привели до появи геометричних GNN, які враховують тривимірні координати атомів, кути зв’язків та просторову конфігурацію молекул, що значно покращує точність відображення їхньої структури. Також було розроблено ланцюгово-орієнтовані архітектури (chain-aware)**,** такі як *MolPath*, які оптимізують передачу повідомлень уздовж

головних ланцюгів молекул, зберігаючи хімічну зв’язаність навіть у великих сполуках.

У контексті відкриття лікарських засобів і прогнозування DDI графові мережі демонструють виняткові результати, оскільки здатні інтегрувати кілька типів даних – хімічну структуру, білкові мішені, ферменти та шляхи метаболізму – в єдині векторні подання. Наприклад:

- моделі ChemProp та HMGNN поєднують інформацію про шляхи та ферменти з графовими ембеддінгами молекул;
- гібридні архітектури GNN–Transformer узгоджують топологію молекулярного графа з біологічними мережами, такими як білок–білкова або лікарський засіб–ензимна взаємодія.

2.2.2 Трансформери для молекулярних представлень

Трансформери стали революційною нейронною архітектурою для навчання молекулярних представлень, забезпечуючи неперевершену гнучкість у моделюванні складних хімічних і біологічних структур. Спочатку розроблені для обробки природної мови, трансформери відзначаються здатністю ефективно виявляти довгострокові залежності та контекстуальні взаємозв’язки в послідовних або структурованих даних завдяки механізму самоуваги (self-attention). У молекулярній сфері цю архітектуру було адаптовано для представлення молекул або як послідовностей (наприклад, рядків SMILES чи SELFIES), або як графів, де атоми виступають вузлами, а хімічні зв’язки - ребрами, що формують елементи для обчислення уваги.

В основі трансформерів лежить операція самоуваги, яка обчислює контекстуальні представлення (embeddings), зважуючи важливість кожного токена відносно інших. Для набору векторів токенів механізм самоуваги обчислюється як:

, (2.17)

де , , , Q (Query) – “запит” для кожного токена,
який шукає релевантну інформацію серед інших токенів, K (Key) – “ключ”, який визначає, яку інформацію кожен токен може надати, V (Value) – і“значення”, тобто вектор, який містить саму інформацію, що передається.
a – розмірність векторів ключів (keys).

Цей механізм дозволяє трансформеру вчитися виявляти контекстно-залежні хімічні зв’язки, такі як атомно-зв’язкова взаємодія або важливість функціональних груп, охоплюючи всю молекулярну структуру, а не лише локальні околиці атомів.

Моделі, що працюють із послідовностями, подібні до мовних моделей: вони токенізують молекули як лінійні рядки (SMILES або SELFIES) і використовують масковане мовне моделювання (Masked Language Modeling, MLM) або автокодування для попереднього навчання.

Такі моделі, як ChemBERTa [36], MolBERT [37] і SMILESTransformer, навчаються на мільярдах неліцензованих молекулярних послідовностей, формуючи приховані вектори, що кодують хімічні закономірності, правила реакцій і семантику підструктур. Ці попередньо навчені представлення потім донавчаються для конкретних завдань: прогнозування властивостей молекул, оцінювання спорідненості “препарат–мішень” або генерації нових сполук *de novo*.

Архітектура encoder–decoder, реалізована в моделях на кшталт DrugAI, розширює цей підхід, поєднуючи навчання представлень із підкріплювальним навчанням (reinforcement learning) та генеративними стратегіями. Наприклад, у DrugAI структура, подібна до BART, генерує нові “drug-like” молекули, вибираючи валідні SMILES, що максимізують кількісні показники подібності до лікарських засобів і передбачену спорідненість

зв’язування. Це демонструє потенціал трансформерів у генеративній хімії та оптимізації кандидатів-лідерів.

Попри успіх у роботі з текстовими послідовностями, трансформери на основі рядків мають обмеження - вони не мають вбудованих 3D індуктивних ознак, що критично важливо для задач, чутливих до геометрії, таких як розпізнавання хіральності або передбачення просторових конформацій.

Графові трансформери долають цю проблему, безпосередньо інтегруючи просторові та топологічні ознаки в механізм уваги. Такі архітектури, як Graphormer і Graphormer-FLAG, збагачують ваги уваги інформацією про центральність вузлів, типи ребер та просторові відстані, дозволяючи моделі враховувати геометрію молекул і структуру зв’язків.

Новітні моделі, як-от Self-Conformation-Aware Graph Transformer (SCAGE), запроваджують багаторівневе конформаційне навчання і багатозадачне попереднє тренування, включно з реконструкцією молекулярних відбитків, класифікацією функціональних груп і передбаченням міжатомних відстаней. Такі підходи дають змогу одночасно сприймати як 2D хімічну зв’язність, так і 3D просторову структуру, ефективно поєднуючи символічні графи з геометричними представленнями молекул.

Сучасні тенденції спрямовані на створення гібридних трансформерних архітектур, які поєднують кілька типів даних – SMILES-послідовності, молекулярні графи, квантово-механічні властивості та біологічну активність – у єдині багатовимірні представлення.

Моделі MolFusion і SMICLR [39], наприклад, реалізують кросмодальне контрастивне навчання (cross-modal contrastive learning), узгоджуючи представлення, отримані з послідовностей і графів, для створення цілісних семантичних ембеддінгів, які краще узагальнюють дані у молекулярних задачах. Такі методи особливо ефективні для прогнозування взаємодій між лікарськими засобами (DDI), де важливо враховувати як молекулярну структуру, так і біохімічний контекст – шляхи, мішені, ферменти тощо.

Трансформери докорінно змінили підходи до прогнозування властивостей молекул, відкриття ліків і моделювання DDI, зробивши можливим масштабоване попереднє навчання на великих хімічних корпусах **і** тонке донавчання на спеціалізованих біомедичних даних.

Ці архітектури забезпечують не лише високу точність прогнозів, а й інтерпретованість результатів завдяки візуалізації уваги, що дозволяє ідентифікувати структурні фрагменти або функціональні групи, найбільш релевантні до активності сполуки чи потенційних лікарських взаємодій.

Отже, трансформери для молекулярних представлень поєднують математичну потужність механізму самоуваги з галузевою специфікою хімічного кодування. Незалежно від того, чи застосовуються вони до послідовностей, графів або мультимодальних поєднань, трансформери забезпечують глибоку, інтерпретовану та гнучку основу для розуміння молекул, генерації нових сполук і прогнозування їхніх взаємодій у фармацевтичних дослідженнях.

2.2.3 Згорткові мережі для SMILES-нотації

Згорткові нейронні мережі (CNN), спочатку розроблені для розпізнавання зображень, були успішно адаптовані для обробки SMILES - символьного лінійного представлення молекулярних графів.

Розглядаючи SMILES як текстову послідовність, CNN можуть ефективно виявляти просторові та контекстуальні закономірності, що дозволяє моделі автоматично вивчати молекулярні підструктури та хімічні мотиви без необхідності у попередньо визначених дескрипторах чи хімічних відбитках (fingerprints).

У моделюванні SMILES на основі CNN послідовність молекули токенізується на атомні символи, типи зв’язків і символи розгалуження, після

чого перетворюється в one-hot або векторне embedding-представлення. Це представлення слугує одновимірним вхідним масивом, до якого застосовуються згорткові фільтри. Згортковий шар із розміром ядра ковзає вздовж послідовності для виявлення локальних закономірностей - комбінацій токенів, що відповідають хімічним підструктурам, таким як ароматичні кільця, атомні групи чи типи зв'язків. Математично операція згортки для вхідного вектора і ядра виражається як:

(2.18)

де – елемент вхідної послідовності (наприклад, токен SMILES); – вагові коефіцієнти ядра згортки (фільтра); – розмір ядра (кількість сусідніх елементів, що враховуються одночасно); – зсув (bias); – нелінійна активаційна функція (найчастіше ReLU); – вихідне значення після застосування фільтра до фрагмента послідовності.

Піонерська робота Hirohara та ін. (2018) [40] продемонструвала, що CNN, навчені на SMILES-представленнях, можуть досягати високої продуктивності в задачах класифікації сполук і прогнозування токсичності. Їхня модель, застосована до еталонного набору TOX21, не лише перевершила традиційні методи на основі відбитків, але й зрівнялася з найкращими хемінформатичними моделями за точністю та інтерпретованістю. Важливо, що візуалізація ознак показала, що фільтри CNN діють як детектори хімічних мотивів, ідентифікуючи як відомі підструктури, так і нові функціональні групи, пов'язані з активністю або токсичністю.

Спираючись на ці результати, подальші дослідження запровадили розширення даних через енумерацію SMILES, визнаючи, що одна молекула може бути представлена кількома еквівалентними SMILES-рядками. Моделі, навчені на розширених наборах SMILES, демонструють підвищену стійкість і узагальненість, оскільки навчаються розпізнавати обертальні та позиційні

інваріантності у різних кодуваннях рядків. Chen і Tseng (2021) [41] ще більше вдосконалили CNN-архітектури для прогнозування розчинності, показавши, що перераховані SMILES-представлення кодують різноманітні структурні конфігурації та створюють хімічно осмислені карти уваги, здатні підкреслювати взаємозв'язки між підструктурами та властивостями.

Сучасні CNN-архітектури, такі як tCNN і CDRScan, об'єднують згорткові шари з attention або щільними шарами, щоб охоплювати глобальні залежності, які втрачаються в суто локальних згорткових системах. Ці моделі успішно застосовуються для прогнозування реакцій на ліки та моделювання механізмів дії препаратів, демонструючи, що згорткові енкодери можуть ефективно перетворювати символічні лінійні нотації у прогностичні молекулярні вектори для фармакологічних завдань.

Для інтеграції CNN з іншими модальностями (наприклад, інформацією про шляхи або білкові цілі) сучасні мультимодальні фреймворки включають паралельні CNN-гілки з молекулярними та біологічними embedding-представленнями. Такі підходи використовують згорткові представлення SMILES як хімічний каркас, пов'язаний із біологічними мережами активності, що забезпечує підвищену інтерпретованість у моделях прогнозування взаємодій між ліками (DDI) або фармакогеномних взаємозв'язків.

Підсумовуючи, згорткові нейронні мережі забезпечують обчислювально ефективний і хімічно виразний метод навчання безпосередньо на SMILES-рядках. Завдяки механізмам згорткового вилучення ознак, ієрархіям pooling та стратегіям гібридного розширення CNN виявляють підструктурні закономірності, що відповідають молекулярним функціям і біологічній активності. Хоча новіші архітектури, такі як трансформери та графові нейронні мережі, розширюють можливості контекстного розуміння, CNN залишаються надійною основою для послідовнісного молекулярного навчання, особливо коли важливими є обчислювальна ефективність і виявлення мотивів.

2.3 Методи інтеграції мультимодальних даних

2.3.1 Early, Late, and Hybrid Fusion

Інтеграція мультимодальних даних є ключовою для сучасних завдань машинного навчання, особливо в біомедичних застосуваннях, де для точних прогнозів необхідно поєднувати таку інформацію, як хімічна структура, біологічні шляхи та фенотипові результати. Ефективність мультимодальних моделей часто залежить від того, як дані з різних модальностей з єднуються між собою; зазвичай це класифікують на три основні підходи: ранню ф'юзію, пізню ф'юзію та гібридну ф'юзію [42].

Рання ф'юзія (feature-level fusion) [43] є одним із найпоширеніших підходів до інтеграції мультимодальних даних, за якого об'єднання інформації відбувається на рівні ознак, ще до етапу моделювання. Це дозволяє моделі навчатися на спільному просторі ознак, що містить низькорівневі кореляції та взаємодії між різними модальностями. Такий метод особливо ефективний тоді, коли між модальностями існують суттєві структурні або семантичні відповідності.

Формальне визначення виглядає наступним чином: допустимо, ми маємо M модальностей, кожна з яких представлена матрицею ознак $X^{(1)}, X^{(2)}, \ldots, X^{(м)}$, $X^{(i)} R^{n \times d}$, де - це кількість зразків, а d - це розмірність ознак модальності . У ранній ф'юзії виконується операція конкатенації:

, (2.19)

де

.

Наступним кроком отриманий простір передається в модель з параметрами , які оптимізуються через мінімізацію функції втрат:

.

Рання ф’юзія є особливо ефективною, коли модальності тісно пов’язані й їхні характеристики можна природно узгодити; однак вона вимагає ретельної попередньої обробки та синхронізації гетерогенних типів даних. Попри потенційну обчислювальну складність через підвищену розмірність, моделі з ранньою ф’юзією здатні захоплювати складні взаємозв’язки, які можуть бути втрачені іншими підходами.

Пізня ф’юзія (late fusion, decision-level fusion) зберігає кожну модальність у відокремленому вигляді протягом усієї обробки та інтегрує їх лише на фінальному етапі шляхом агрегування незалежних прогнозів або виходів окремих унімодальних моделей. Цей підхід є високорозбірним (модульним), що полегшує додавання або вилучення модальностей без потреби повного перенавчання системи. Що в свою чергу надає системі високу модульність, стійкість до пропусків у даних та гнучкість щодо додавання чи вилучення модальностей. На відміну від ранньої ф’юзії, тут моделі не мають спільного простору ознак – кожна працює автономно.

Формальне визначення виглядає наступним чином: нехай маємо M модальностей, кожну з яких опрацьовує окрема модель для вибірки кожної модальності , $i = 1,2,\ldots,M$.

Мета late fusion – побудувати агрегуючу функцію

.

Фукнція F може бути: простою (середнє значення), зваженою, експертною (majority voting), навченоб (meta-learned).

Основними перевагою цього підходу, як вже було зазначено вище: модульність (моделі незалежно навчаються на своїх наборах даних), стійкість до пропусків (якщо модальність пропущена, то все ще працює) та низька вимога до узгодження даних, на відміну від ранньої ф'юзії [44].

Гібридна ф'юзія прагне поєднати сильні сторони ранньої та пізньої ф'юзії, інтегруючи модальності на кількох рівнях моделі. Це підхід до мультимодальної інтеграції, який поєднує елементи ранньої (feature-level) та пізньої (decision-level) ф'юзії. Іншими словами, дані частково інтегруються на рівні ознак, а частково - на рівні прогнозів або високорівневих репрезентацій. Завдяки цьому модель може одночасно вловлювати і низькорівневі взаємодії між ознаками (як у early fusion), і високорівневі міжмодальні залежності (як у late fusion). Цей підхід особливо успішний у складних задачах, де лише рання або лише пізня ф'юзія дає неповне представлення мультимодальної природи даних.

Формальне визначення виглядає наступним чином: допустимо, ми маємо M модальностей, кожна з яких представлена матрицею ознак $X^{(1)}, X^{(2)}, \ldots, X^{(м)}$. Для кожної модальності визначимо два рівні представлення:

, (2.20)

де – функція екстракції ознак з модальності ; це може бути CNN, GNN або ж autoencoders, - низькорівневе (feature-level) представлення модальності .

, (2.21)

де – предикативна модель для модальності , – високорівневий вихід моделі , зазвичай близький до прогнозу.

Гібридна фузія формально комбінує ранню і пізню:

, (2.22)

де – оператор feature-level fusion**,** – оператор decision-level fusion, – фінальний агрегатор, що інтегрує обидві ф’юзії.

2.3.2 Механізми уваги

Механізми уваги (Attention Mechanisms) слугують потужною обчислювальною парадигмою для інтеграції та зважування ознак, що дозволяє нейронним мережам динамічно фокусуватися на найбільш інформативних компонентах вхідних даних. На відміну від традиційних підходів, які трактують усі вхідні дані як рівнозначні, увага призначає навчувані коефіцієнти важливості (Attention Weights). Це дає змогу моделі підсилювати релевантні молекулярні підструктури, біологічні мішені, ферменти або сигнальні шляхи залежно від контексту конкретного прогностичного завдання. Концептуально, увага функціонує як диференційований селекційний процес: вона підсилює суттєві сигнали та пригнічує шум, що критично покращує як точність прогнозів, так і інтерпретованість результатів у складних біомедичних доменах.

Історично, механізми уваги набули широкого визнання після їхньої інтеграції в архітектуру Transformer (Vaswani et al., 2017), яка замінила рекурентні (RNN) та згорткові (CNN) мережі в завданнях обробки природної мови (NLP), а згодом була адаптована для молекулярних та біомедичних даних.

Механізми уваги забезпечують потужну основу для інтеграції мультимодальних даних, дозволяючи моделям динамічно фокусуватися на найбільш інформативних ознаках як між модальностями, так і всередині кожної модальності. Замість того щоб розглядати всі вхідні дані рівнозначними, механізм уваги призначає навчені вагові коефіцієнти важливості, завдяки яким мережа може підсилювати релевантні молекулярні підструктури, мішені, ферменти або сигнальні шляхи залежно від контексту прогнозу. Концептуально увагу можна розглядати як диференційований селекційний процес, який підсилює суттєві сигнали та пригнічує шум, покращуючи як точність прогнозів, так і інтерпретованість у складних біомедичних завданнях.

У математичному формулюванні поширеним варіантом є scaled dot-product attention. Нехай задано вектор запиту (q, K, V). Тоді увага обчислюється як зважена комбінація значень:

$$, \qquad (2.23)$$

де q – матриця запитів, K – матриця ключів, V – матриця значень, d_k – розмірність ключів.

Оператор softmax формує розподіл ймовірностей над ключами, що відображає, наскільки сильно запит «уважує» до кожного елемента. У мультимодальних налаштуваннях запити, ключі та значення можуть походити з різних модальностей - наприклад, коли SMILES-вектор лікарського засобу використовується як запит до ембедингів шляхів або мішеней, що явно моделює крос-модальні взаємозв’язки.

Self-attention - це механізм, який дозволяє *моделі* динамічно визначати важливість різних елементів усередині однієї й тієї ж модальності. На відміну від традиційних моделей, де кожний елемент обробляється ізольовано або за фіксованою послідовністю, self-attention дає змогу кожному елементу

«дивитися» на всі інші елементи того самого набору та переважувати їх залежно від завдання.

У контексті молекулярних даних self-attention дозволяє моделі дізнатися які атоми або підструктури найбільш критичні для активності, як локальні фрагменти пов’язані з глобальними властивостями та які взаємодії вказують на специфічні механізми дії або токсичність.

Нехай маємо матрицю ознак , де це кількість

елементів (атомів, токенів SMILES, нод графа), а це розмірність ознак.

У self-attention *з* тих самих даних конструюються: $Q=XW_Q$, $K=XW_K$, $V=XW_V$, де - навчувані матриці проєкцій; - матриця запитів; - матриця ключів; - матриця значень.

, (2.24)

де – подібність між кожним елементом (запитом) та всіма елементами (ключами); – масштабування, що запобігає занадто великим значенням;

- нормалізує ваги для кожного елемента, утворюючи розподіл уваги; V – формує зважене представлення, акцентоване на важливих елементах.

Хоча self-attention працює в межах однієї модальності, у мультимодальних архітектурах він виконує важливі функції. Перед тим як SMILES взаємодіє з шляхами (pathways) або таргетами self-attention робить представлення хімічної структури чистішим і структурно-узгодженим. Для хімії це мотиваційні групи, фармакофорні елементи, ключові субграфи. У біологічних профілях (наприклад, експресія генів): self-attention підсилює

гени/шляхи, які релевантні прогнозу, пригнічує слабкосигнальні або нерелевантні гени.

Multi-head self-attention (MHSA) є розширенням класичного механізму self-attention, яке дозволяє моделі одночасно захоплювати кілька різних типів залежностей між елементами однієї й тієї самої модальності. Нехай вхідна матриця ознак має вигляд , де - кількість елементів, а - розмірність ознак. Для кожної з голів самоуваги формуються окремі запити, ключі та значення через навчувані матриці проєкцій для , такі що

. (2.24)

Кожна голова обчислює зважену комбінацію значень через scaled dot-product attention:

head softmax (2.26)

Після обчислення всіх голів їхні виходи конкатенуються, формуючи інтегроване представлення:

MHSA concat head head (2.27)

де – навчувана матриця проєкції для об’єднання результатів усіх голів у простір початкової розмірності . Така архітектура дозволяє одночасно захоплювати локальні та глобальні залежності, різні аспекти структури та взаємодії між елементами, що особливо важливо у молекулярному машинному навчанні для SMILES, молекулярних графів або біологічних послідовностей. Кожна голова може навчитися виділяти специфічні патерни, наприклад коротко- або довготривалі залежності,

структурні мотиви чи функціональні групи, і їхнє паралельне комбінування забезпечує багатовимірне, багатоперспективне представлення даних, що підвищує точність прогнозування та інтерпретованість моделі.

В свою чергу Cross-attention це механізм уваги, який дозволяє інтегрувати інформацію з різних модальностей або джерел даних, забезпечуючи динамічне зважування значущості елементів однієї модальності щодо іншої.

Нехай - матриця ознак модальності-запиту (query), де - кількість елементів, а - розмірність ознак. Нехай - матриця ознак модальності-ключ/значення (key/value), де - кількість елементів, а - розмірність ознак.

Для обчислення cross-attention формуються запити , ключі та значення за допомогою навчуваних матриць проєкцій , та , де - розмірність простору ключів/запитів, а - розмірність простору значень:

(2.28)

де – матриця запитів; – матриця ключів; – матриця значень; – параметри, які мережа навчає під час тренування, щоб оптимально перетворювати вхідні ознаки.

Вихід cross-attention обчислюється як:

CrossAttention softmax (2.29)

де – матриця схожості між кожним елементом запиту та ключами; – масштабуючий фактор, який запобігає надто великим значенням у softmax; softmax – нормалізує значення вздовж рядків,

формуючи розподіл уваги для кожного запиту; добуток на дає зважене поєднання значень, де найбільша увага приділяється найбільш релевантним елементам ключів.

Таким чином, кожен елемент модальності-запиту отримує адаптивне, контекстуально зважене представлення на основі інформації з іншої модальності. У мультимодальних задачах, таких як передбачення взаємодій лікарських засобів (DDI), cross-attention дозволяє пов’язати структурні представлення молекули (SMILES або граф) з біологічними даними мішеней, ферментів або шляхів, забезпечуючи одночасно високу точність прогнозів та інтерпретованість моделі.

Механізми уваги, включаючи self-attention, cross-attention та їх багатоголові варіанти, забезпечують потужний інструментарій для інтеграції та аналізу мультимодальних даних у задачах молекулярного машинного навчання. Вони дозволяють моделі динамічно зважувати релевантність окремих елементів усередині однієї модальності або між різними модальностями, тим самим створюючи контекстуально адаптивні представлення. У мультимодальному передбаченні взаємодій лікарських засобів (DDI) це означає, що модель може одночасно захоплювати різноманітні патерни взаємодій між SMILES-ембедингами, профілями мішеней, ферментативною інформацією та участю у сигнальних шляхах. Self-attention дозволяє підсилювати критично важливі атоми, підструктури чи компоненти графів, тоді як cross-attention забезпечує ефективну інтеграцію інформації між хімічними та біологічними модальностями, дозволяючи встановлювати прямі відповідності між структурними характеристиками молекул і функціональними чи біологічними контекстами. Багатоголовий механізм уваги забезпечує одночасне вивчення кількох аспектів залежностей, таких як локальні й глобальні взаємозв’язки, структурні мотиви та функціональні групи, що сприяє більш насиченому та гнучкому представленню даних. Завдяки цим властивостям attention-базована інтеграція є не лише ефективним способом ф’юзії різнорідних даних, але й

забезпечує високу інтерпретованість моделей, що особливо важливо для біомедичних та хіміко-біологічних застосувань. Таким чином, механізми уваги формують концептуальне ядро сучасних трансформерних та гібридних архітектур у молекулярному машинному навчанні, забезпечуючи одночасно точність прогнозування, адаптивність до складних мультимодальних структур та прозорість для аналізу результатів моделі.

2.2.3 Крос-модальне навчання

Cross-modal learning [45] є передовою парадигмою інтеграції мультимодальних даних, де моделі навчаються відтворювати відповідності та вирівнювання між різними модальностями для отримання спільних латентних представлень. На відміну від традиційної ф’юзії, яка лише конкатенує ознаки, cross-modal підходи навчають мережі транслювати інформацію з однієї модальності в простір іншої. Наприклад, структурне представлення лікарської молекули SMILES може бути відображене у простір біологічних мішеней або сигнальних шляхів. Це забезпечує двосторонній перенос знань та дозволяє виявляти відповідності між хімічними властивостями та біологічною функцією, що особливо важливо для передбачення складних взаємодій лікарських засобів (DDI).

У cross-modal learning часто використовують контрастивні цілі, щоб вирівняти ембединги різних модальностей. Нехай - ембединг елемента з однієї модальності (наприклад, SMILES), а - відповідний ембединг з іншої модальності (наприклад, ембединг мішені). Контрастивна функція втрат визначається як:

$$\text{contrastive} \quad \frac{\text{sim}}{\text{sim}} \tag{2.30}$$

де – ембедінги відповідних елементів із різних модальностей, – ембедінги інших елементів у батчі (негативні приклади), sim – метрика подібності, наприклад косинусна схожість: sim ,

розмір батчу, – температурний параметр, який масштабує значення перед softmax, – експоненціальна функція, що перетворює схожість у позитивні значення; знаменник сумує внески всіх елементів батчу, формуючи нормалізований розподіл уваги для позитивної пари.

Цей підхід змушує позитивні пари наближатися в латентному просторі, тоді як негативні - відштовхуються, що дозволяє моделі навчатися модально-незалежних, фармакологічно релевантних ознак.

Encoder-decoder архітектура в свою ж чергу концептуально відрізняється від попереднього підходу. Нехай – енкодер для модальності (наприклад SMILES), а – енкодер для модальності (наприклад мішені). Вони генерують ембедінги:

(2.31)

де та – кількість елементів у відповідних модальностях, – розмірність ембедингів.

Декодер може відтворювати одну модальність з ембедингів іншої або передбачати мітки взаємодій:

(2.32)

де – реконструйовані або передбачені представлення, а - розмірності початкових ознак.

Функція втрат для реконструкції може бути визначена як середньоквадратична помилка (MSE):

recon – — (2.33)

Для більш ефективного поєднання модальностей використовують механізм cross-attention. Нехай – запити з модальності SMILES, , – ключі та значення з модальності мішеней, де – навчувані матриці проєкцій. Тоді крос-модальна увага обчислюється як:

CrossAttention softmax (2.34)

де , – ембедінги модальностей, , , – запити, ключі та значення, – розмірності ключів і значень, softmax – нормалізація, яка визначає ваги уваги для кожного елемента запиту, добуток на формує зважене представлення, де увага приділяється найбільш релевантним елементам ключів.

Це дозволяє кожному елементу SMILES отримувати контекстуально збагачену інформацію про біологічні мішені, ферменти або сигнальні шляхи, що підвищує точність передбачення взаємодій.

У практичних архітектурах cross-modal learning часто комбінують контрастивне препрограмування з task-specific fine-tuning. Контрастивне навчання забезпечує модально-незалежні ембедінги, а cross-attention інтегрує їх для конкретної задачі DDI. Це дозволяє моделі передбачати взаємодії

навіть для нових комбінацій лікарських засобів, де окремі модальності надають неповну інформацію.

На практиці кросмодальне навчання застосовується для інтеграції гетерогенних даних, таких як SMILES-представлення молекул, біологічні мішені, ферменти та сигнальні шляхи. Незважаючи на спільну мету - навчитися узгоджувати інформацію між різними модальностями - підходи суттєво відрізняються за механізмом навчання, обмеженнями та практичною ефективністю.

Контрастивне навчання (contrastive learning) орієнтоване на створення спільного латентного простору, де позитивні пари (елементи з різних модальностей, які відповідають одній і тій самій сутності) притягуються, а негативні пари відштовхуються. У практичних застосуваннях для DDI це дозволяє моделі виявляти узагальнені відповідності між SMILES-підструктурами та профілями мішеней або шляхів. Головна перевага цього підходу полягає у здатності моделі узагальнювати знання для нових комбінацій лікарських засобів, навіть якщо конкретні приклади в навчальному наборі відсутні. Контрастивне навчання добре масштабується на великі датасети і забезпечує стійкість до неповних або частково відсутніх модальностей. Водночас його основний недолік полягає в тому, що модель навчається виявляти подібності, але не обов’язково створює пряме відтворення однієї модальності через іншу, що обмежує її здатність до деталізованої інтерпретації конкретних структурно-біологічних взаємодій.

Реконструкційне навчання (reconstruction-based learning), навпаки, передбачає навчання моделей відтворювати одну модальність з іншої, використовуючи encoder-decoder архітектуру. Наприклад, SMILES-енткодер формує ембединги молекули, а декодер реконструює профіль мішеней або шляхи, що дозволяє моделі вчитися прямим функціонально-структурним відповідностям. Практична користь такого підходу полягає у високій інтерпретованості та здатності моделі пояснювати свої передбачення: можна визначити, які підструктури молекули відповідають за певні біологічні

властивості. Недоліком є висока чутливість до якості даних: якщо одна з модальностей містить шум або неповні дані, реконструкція стає неточною, що негативно впливає на продуктивність моделі. Крім того, такі архітектури потребують більшої кількості параметрів та обчислювальних ресурсів через наявність декодера для кожної модальності.

Attention-based cross-modal моделі, які використовують cross-attention або multi-head attention, поєднують переваги обох підходів. Вони дозволяють динамічно інтегрувати інформацію між модальностями на рівні окремих елементів**,** де запити однієї модальності звертаються до ключів та значень іншої модальності. На практиці це забезпечує моделі здатність одночасно захоплювати локальні та глобальні залежності: наприклад, визначати, як конкретна SMILES-підструктура впливає на активність ферменту чи участь у сигнальних шляхах. Cross-attention підвищує точність прогнозування DDI навіть у випадках, коли комбінації молекул не були представлені у навчальному наборі. Основні обмеження цього підходу полягають у підвищеній обчислювальній складності та потребі в ретельному налаштуванні attention-механізмів для різних модальностей, щоб уникнути перенавчання або надмірного приділення уваги одній модальності за рахунок іншої.

Механізми кросмодального навчання та attention становлять фундаментальну основу сучасних підходів до інтеграції мультимодальних даних у молекулярному машинному навчанні та передбаченні взаємодій лікарських засобів (DDI). Розглянуті підходи - контрастивне навчання, реконструкційне навчання та attention-based інтеграція - відрізняються не лише архітектурною реалізацією, а й практичними можливостями та обмеженнями. Контрастивне навчання дозволяє моделі створювати спільний латентний простір, де позитивні пари з різних модальностей притягуються, а негативні - відштовхуються, що забезпечує ефективне узагальнення знань та стійкість до неповних або шумних даних. Реконструкційне навчання, навпаки, акцентує на здатності моделі безпосередньо відтворювати одну

модальність на основі іншої, що підвищує інтерпретованість і дозволяє досліджувати конкретні відповідності між структурними та біологічними ознаками. Attention-based моделі, особливо ті, що використовують cross-attention і multi-head attention, забезпечують динамічну інтеграцію інформації між модальностями на рівні окремих елементів, дозволяючи одночасно захоплювати локальні та глобальні залежності та формувати контекстуально збагачені представлення.

У практичному застосуванні до передбачення DDI ці підходи взаємодоповнюють один одного. Контрастивне навчання сприяє формуванню модально-незалежних ембедингів, які ефективно узагальнюють знання про нові комбінації лікарських засобів. Реконструкційне навчання забезпечує прозорі зв’язки між молекулярною структурою, біологічними мішенями та шляхами метаболізму, що сприяє інтерпретованості моделі. Attention-механізми дозволяють моделі визначати «куди дивитися» в гетерогенних даних, динамічно зважуючи інформацію з різних модальностей та інтегруючи її в багатовимірні представлення, що відображають складні взаємодії між хімічними, біологічними та системними властивостями молекул.

Таким чином, кросмодальне навчання та attention-базовані механізми формують гнучкий, інтерпретований та високотехнологічний підхід до мультимодальної інтеграції**,** який дозволяє моделі одночасно враховувати структурні, функціональні та системні аспекти біології лікарських засобів. Вибір конкретного підходу залежить від цілей дослідження, обсягу та якості даних, а також від потреби у точності та інтерпретованості прогнозів. У комплексі ці методи забезпечують підвищену продуктивність моделей, їхню генералізацію на нові комбінації молекул та формують основу для створення інтерпретованих і надійних фармакоінформатичних систем.

Висновки до розділу 2

У другому розділі систематизовано теоретичні та методологічні основи, необхідні для проєктування інтелектуальної системи прогнозування взаємодій між лікарськими засобами на базі мультимодальних даних. Розділ охоплює математичні основи глибокого навчання, спеціалізовані архітектури для обробки молекулярних даних та методи інтеграції мультимодальної інформації.

У підрозділі 2.1 встановлено математичний фундамент глибинного навчання. Розглянуто базову структуру нейронних мереж, включаючи багатошарові перцептрони та принцип універсальної апроксимації. Проаналізовано функції активації (ReLU, LeakyReLU, Sigmoid, Tanh, Swish), методи оптимізації (SGD, Adam, AdaGrad, RMSprop) та техніки регуляризації (Dropout, Batch Normalization, L2- регуляризація), які забезпечують ефективне навчання та запобігають перенавчанню. Охарактеризовано функції втрат для різних типів завдань: MSE та MAE для регресії, BCE для бінарної класифікації, CCE для багатокласової класифікації та Focal Loss для роботи з дисбалансом класів.

Підрозділ 2.2 присвячено архітектурам для обробки молекулярних даних. Графові нейронні мережі (GNN) представлено як природний інструмент для обробки молекул через механізми message passing та агрегації вузлів, що ефективно захоплюють топологічні властивості. Трансформери (ChemBERTa, MAT, Graphormer) розглянуто як революційний підхід з механізмом self-attention, що дозволяє динамічно фокусуватися на релевантних молекулярних фрагментах та встановлювати далекодіючі залежності. Згорткові мережі для SMILES-нотації представлено як обчислювально ефективний метод виявлення локальних хімічних мотивів через згорткові фільтри. У підрозділі 2.3 систематизовано методи інтеграції мультимодальних даних. Розглянуто три стратегії злиття: ранню ф'юзію

(інтеграція на рівні ознак), пізню ф'юзію (об'єднання незалежних прогнозів) та гібридну ф'юзію (комбінація обох підходів). Особливу увагу приділено механізмам уваги (scaled dot-product attention, self-attention, multi-head attention, cross-attention), що дозволяють динамічно зважувати важливість модальностей через матриці запитів, ключів та значень. Розглянуто крос-модальне навчання для встановлення відповідностей між хімічною структурою, білковими мішенями, ферментами та біологічними шляхами. Теоретичний апарат розділу створює фундамент для проєктування DDI-TransMDL. Обґрунтовано вибір архітектури трансформер-енкодера як оптимального рішення для міжмодальної інтеграції завдяки multi-head self-attention, що вивчає нелінійні залежності між модальностями. Методи регуляризації та оптимізації забезпечують стратегії ефективного навчання на обмежених біомедичних даних. Систематизація архітектур (GNN, трансформери, CNN) надає інструментарій для проєктування спеціалізованих енкодерів для кожної модальності SMILES, мішеней, ферментів та шляхів.

Розділ демонструє еволюцію від одномодальних підходів до мультимодальних архітектур, відображаючи розуміння взаємодій ліків як мультифакторного явища. Математична формалізація всіх методів забезпечує строге обґрунтування та можливість практичної реалізації. Встановлено зв'язок між загальними принципами глибокого навчання та специфічними потребами молекулярного моделювання, показуючи необхідність адаптації архітектур до гетерогенної природи даних та вимог інтерпретованості.

Таким чином, розділ забезпечує всебічну теоретичну підготовку для створення інтелектуальної системи прогнозування DDI, поєднуючи математичну строгість, сучасні досягнення глибокого навчання та специфічні вимоги молекулярного моделювання, створюючи надійну основу для проєктування, реалізації та валідації системи DDI-TransMDL.

РОЗДІЛ 3 ПРОЄКТУВАННЯ МУЛЬТИМОДАЛЬНОЇ МОДЕЛІ ПРОГНОЗУВАННЯ ВЗАЄМОДІЇ ЛІКІВ (DDI-TRANSMDL)

Прогнозування взаємодії лікарських засобів (DDI) є критично важливою задачею у фармакології та клінічній медицині, оскільки непередбачувані DDI можуть призводити до серйозних побічних реакцій або зниження ефективності лікування. Розробка обчислювальної моделі для цієї задачі вимагає ефективного механізму для інтеграції гетерогенних даних, які описують хімічні властивості, біологічну активність та метаболічні шляхи ліків. Традиційні методи, засновані на простих метриках подібності або класичних алгоритмах машинного навчання, часто не здатні захопити комплексні нелінійні взаємозв'язки, що лежать в основі DDI.

Цей розділ присвячений проєктуванню моделі DDI-TransMDL (Transformer-based Drug-Drug Interaction Prediction Model), яка використовує архітектуру Трансформер-Енкодера для вирішення цієї проблеми.

Основна мета роботи полягає у розробці та впровадженні високоефективної моделі на основі архітектури Трансформера для прогнозування потенційно небезпечних взаємодій між лікарськими препаратами (DDI), використовуючи мультимодальні дані.

Враховуючи, що DDI можуть призводити до серйозних побічних ефектів, задача формулюється як проблема багатоміткової класифікації (Multi-label Classification), що дозволяє не лише прогнозувати факт наявності взаємодії, але й ідентифікувати її конкретний тип або механізм.

Вхідні дані для моделі X є парою лікарських препаратів (D_A, D_B) представленою через конкатенацію чотирьох незалежних векторів ознак (модальностей). Де кожен вектор має фіксовану довжину 1144 і включає конкатенацію ознак препарату D_A, ознак препарату D_B та коефіцієнта подібності Жаккарда між ними для відповідної модальності.

Вихідні дані Y являють собою бінарний вектор прогнозу, де кожен елемент відповідає одній із 65 унікальних DDI-подій або типів взаємодії, ідентифікованих у базі даних. Де всі елементи векторі {0, 1}. Значення вказує на те, що прогнозується наявність -ї події взаємодії (наприклад, "збільшення концентрації вільного препарату в плазмі"), а
- на її відсутність.

3.1 Обґрунтування вибору архітектури трансформера

Проєктування ефективної обчислювальної моделі для прогнозування взаємодії ліків (DDI) вимагає архітектурного рішення, здатного до глибокої та гнучкої інтеграції мультимодальних даних. Джерела інформації про ліки, такі як хімічна структура, білкові мішені, ферменти та біологічні шляхи, є гетерогенними і вимагають механізму, що здатен динамічно оцінювати їхній внесок у фінальний прогноз. Традиційні архітектури, що використовують послідовну конкатенацію або просте усереднення ознак, часто виявляються недостатньо чутливими до тонких, нелінійних залежностей, які виникають при взаємодії двох активних фармацевтичних інгредієнтів.

У цьому контексті, архітектура Трансформера була обрана як фундаментальна основа DDI-TransMDL. Вона є найбільш придатною для вирішення задачі міжмодальної інтеграції, завдяки її ключовому компоненту - механізму Multi-Head Self-Attention. Цей механізм дозволяє моделі не просто об'єднувати ознаки, а й вивчати складні залежності та виявляти, наскільки ознака однієї модальності (наприклад, фермент) є релевантною для ознак іншої (наприклад, цільовий білок) при прогнозуванні конкретної DDI-події. Такий підхід забезпечує високу інтерпретованість та обчислювальну

ефективність, що є критично важливим для створення надійного та масштабованого інструменту прогнозування.

3.1.1 Вимоги до моделі для прогнозування DDI

Проєктування моделі DDI-TransMDL для прогнозування взаємодії між лікарськими засобами вимагає врахування специфічних характеристик вхідних даних та природи самої задачі. Успішне вирішення задачі DDI-прогнозування залежить від здатності моделі відповідати трьом ключовим методологічним вимогам: обробка гетерогенних даних, виявлення комплексних нелінійних залежностей та глибока інтеграція ознак пари ліків.

Модель повинна інтегрувати інформацію з різних біологічних, хімічних та фармакологічних доменів. Це вимагає, щоб архітектура була спроєктована для прийому чотирьох незалежних наборів векторів ознак, що представляють пару ліків (D_A, D_B).

Початковий вхідний простір формалізується як набір векторів ознак, де кожен вектор V_M відповідає певній модальності M для пари (D_A, D_B):

, (3.1)

де – загальний вхідний набір даних для моделі, – два лікарські засоби, що утворюють пару, – множина ключових модальностей, визначена як = {SMILE, Target, Enzyme, Pathway}, – вектор ознак, що описує пару виключно з точки зору модальності .

У проєкті висувається вимога, що кожен вектор ознак пари матиме уніфіковану фінальну розмірність = 1144.

Взаємодія ліків є результатом складних біологічних каскадів, що вимагає, щоб модель вміла моделювати глибокі нелінійні зв'язки. Це особливо стосується міжмодальних залежностей - впливу ознак однієї модальності на ознаки іншої.

Обрана архітектура повинна забезпечити обчислення зваженої взаємодії між модальністю M_i і модальністю M_j для прогнозування конкретної DDI-події. Це реалізується за допомогою механізму уваги (Attention), який динамічно генерує вагу

, (3.2)

де – вбудовування (embedding) модальностей M_i та M_j у просторі , – навчені вагові матриці, що визначають функції запиту, ключа та значення (Query, Key, Value), – коефіцієнт уваги, який показує важливість модальності j для обробки модальності і в поточному контексті.

Завдяки цьому, вимога до моделі - забезпечити, щоб фінальне представлення агрегувало інформацію з урахуванням цих нелінійних і зважених міжмодальних залежностей.

Модель повинна не тільки обробляти модальності, але й ефективно інтегрувати інформацію про два ліки та прогнозувати багатомітковий вихід.

Для посилення інформації про взаємодію на рівні входу, проєктом вимагається явне включення метрики подібності. Використовується коефіцієнт Жаккарда , як метрика спільності для кожної модальності *M*.

, (3.3)

де – кінцевий вектор ознак модальності M для пари, – вектори ознак ліків та 572 кожен, –

коефіцієнт Жаккарда, що вимірює спільність дискретних ознак (наприклад, цільових білків) між двома ліками.

Вимога полягає у тому, щоб (розмірністю 1144) містив це явне скалярне значення, інформуючи Трансформер про потенційний конфлікт або синергію вже на вхідному етапі.

Модель повинна вирішувати задачу багатоміткової бінарної класифікації (Multi-label Classification), прогнозуючи наявність чи відсутність кожної з C можливих DDI-подій.

, (3.4)

де – прогнозований вектор ймовірностей, C=65 (кількість DDI-подій), – фінальний агрегований вектор, отриманий з токена [CLS] після проходження трансформера, – функція класифікаційної голови (стек лінійних шарів), – функція активації, що застосовується до кожного з C виходів незалежно, гарантуючи, що є ймовірністю належності до події c.

Для навчання цієї системи потрібна функція втрат Binary Cross-Entropy Loss.

Ключова проблема, яку вирішує проєкт, полягає в обробці гетерогенних мультимодальних даних. Модальності SMILE, Target, Enzyme та Pathway, які мають різну семантичну природу, вимагають не простого об'єднання, а уніфікації та перетворення у послідовність токенів. Це концептуальне перетворення дозволяє застосувати архітектуру Трансформер-Енкодера, де кожна модальність розглядається як незалежний токен у послідовності.

Центральна вимога до функціональності моделі - глибока міжмодальна інтеграція. У традиційних моделях взаємодія між, наприклад, хімічною структурою (SMILE) та метаболічною активністю (Enzyme) є неявною.

Проєктування ж моделі DDI-TransMDL, що використовує механізм Multi-Head Self-Attention, дозволяє розрахувати динамічні вагові коефіцієнти між усіма парами модальностей. Цей механізм уваги забезпечує найвищий ступінь нелінійної взаємодії та дає змогу моделі виявляти складні залежності.

Крім того, проєктом вимагається явна інтеграція ознак взаємодії. Просте конкатенування векторів ознак ліків та недостатнє. Для цього була введена метрика коефіцієнта Жаккарда , яка додається безпосередньо до вхідного вектора (розмірність 1144).
Це гарантує, що Трансформер обробляє як індивідуальні профілі ліків, так і метрику їхньої спільності в єдиному контексті уваги.

Нарешті, фінальна вимога - це багатомітковий прогноз. Завдяки агрегації інформації в токені та застосуванню класифікаційної голови з функцією Sigmoid та функцією втрат , модель повністю відповідає вимогам до багатоміткової класифікації, здатна незалежно прогнозувати ймовірність кожної із 65 DDI-подій.

Таким чином, архітектура Трансформер-Енкодера обрана не просто як глибока нейронна мережа, а як спеціалізований механізм для зваженої інтеграції послідовності гетерогенних токенів, що є вирішальним для точного моделювання складних біологічних взаємозв'язків DDI.

3.1.2 Обґрунтування вибору архітектури трансформер-енкодера

Вибір архітектури Трансформер-Енкодера замість традиційних нейронних мереж, таких як згорткові (CNN), рекурентні (RNN) або графові (GNN) моделі, є ключовим елементом у проєктуванні DDI-TransMDL. Це рішення продиктоване необхідністю ефективної інтеграції гетерогенних, але

фіксованих за кількістю модальностей, а також потребою у моделюванні їхніх складних взаємозв’язків у єдиному обчислювальному просторі.

На відміну від рекурентних архітектур, що опрацьовують дані послідовно, Трансформер забезпечує негайне захоплення залежностей між будь-якими елементами вхідної послідовності, що усуває характерну для RNN проблему довгострокових залежностей. Хоча послідовність модальностей у DDI-TransMDL є короткою і має фіксовану довжину п’ять токенів, механізм самоуваги дозволяє миттєво встановлювати зв’язки між віддаленими токенами, наприклад між SMILE та Pathway. Паралельна обробка, притаманна Трансформеру, істотно прискорює навчання, забезпечуючи кращу масштабованість у порівнянні з RNN.

Згорткові нейронні мережі, попри високу ефективність у виявленні локальних просторових закономірностей, мають обмежену здатність до формування глобальної рецептивності. Для досягнення глобального погляду CNN потребують значної кількості згорткових шарів, що ускладнює модель і збільшує обчислювальні витрати. У завданнях прогнозування DDI критично важливою є глобальна взаємодія між модальностями: токен структурного представлення молекули повинен мати можливість взаємодіяти з токенами, що відповідають за ферментативну активність або участь у шляхах. Саме механізм самоуваги надає змогу токену $_{SMILE}$ безпосередньо «звертатися» до токена $_{Enzym}$ незалежно від їхнього розташування у послідовності.

Центральною властивістю, що визначає переваги Трансформера, є механізм Multi-Head Self-Attention. Він реалізує динамічну агрегацію внеску кожної модальності, дозволяючи моделі адаптивно визначати ступінь важливості кожного токена залежно від контексту конкретної DDI-події. Формально це виражається через обчислення оновленого представлення токена.

(3.5)

де – довжина послідовності модальностей, – вектор значення токена , – коефіцієнт уваги, що визначає силу впливу токена на токен .

Завдяки використанню восьми незалежних голів уваги модель здатна захоплювати різні типи взаємозв’язків між модальностями. Одні голови можуть спеціалізуватися на структурних відповідностях між SMILES-представленнями, тоді як інші - на виявленні фармакокінетичних взаємодій між ферментами та мішенями. Така багатоголова структура забезпечує глибоку та різнопланову міжмодальну інтеграцію.

Графові нейронні мережі (GNN) є ефективними для задач, де критичною є складна топологічна структура та нерегулярні графові взаємодії, наприклад, моделювання великих біологічних мереж. Проте у завданні прогнозування DDI кількість модальностей є сталою та невеликою, що зводить проблему до зваженої інтеграції ознак, а не до моделювання складної графової структури. У цьому випадку Трансформер-Енкодер виступає природною альтернативою, оскільки сам механізм уваги інтерпретується як повний граф, де кожен токен може взаємодіяти з будь-яким іншим токеном через пряме зважене ребро. Це робить архітектуру Трансформера ефективним засобом моделювання інтенсивних міжмодальних взаємодій без необхідності явного побудування графової топології.

Ключова роль механізму Self-Attention у проєкті DDI-TransMDL полягає у створенні динамічного, контекстно-залежного агрегатора для чотирьох гетерогенних модальностей: SMILE, Target, Enzyme та Pathway. На відміну від простих зважених сум або конкатенації, Self-Attention дозволяє моделі не просто об'єднувати, а й зважувати значущість кожної модальності відносно всіх інших у послідовності.

Механізм Self-Attention у Трансформер-Енкодері забезпечує, що кожен токен (модальність) у послідовності має прямий, неієрархічний зв'язок з усіма іншими токенами. Це особливо важливо, оскільки в DDI-прогнозуванні критичні взаємозв'язки можуть існувати між будь-якою парою модальностей,

наприклад, між хімічною структурою $_{SMILE}$ та метаболічними ферментами $_{Enzym}$•

Для кожного токена E_i (що представляє певну модальність), механізм обчислює коефіцієнти уваги щодо всіх інших токенів E_j. Це обчислення базується на трьох лінійних проєкціях вхідного токена: Запит (Query, Q), Ключ (Key, K) та Значення (Value, V) (2.23).

На відміну від статичних методів злиття ознак, Self-Attention дозволяє динамічне зважування модальностей. Наприклад, якщо пара ліків взаємодіє через спільну активацію метаболічного шляху, токени $_{Pathway}$ та $_{Enzym}$ отримають високі коефіцієнти уваги , домінуючи у фінальному представленні . Якщо ж DDI спричинена структурним конфліктом, увага буде зосереджена на $_{SMILE}$. Це гарантує, що модель адаптується до конкретного механізму взаємодії для кожної пари.

Застосування механізму Multi-Head Self-Attention з N_{HEADS}=8 головами є критичним для виявлення комплексних взаємозв'язків. Кожна з восьми незалежних голів уваги виконує операцію Attention паралельно, використовуючи різні навчальні вагові матриці W^Q, W^K , W^V. Це дозволяє моделі одночасно моделювати різні аспекти міжмодальних відносин: одна голова може виявляти фармакокінетичні взаємодії (наприклад, $_{Enzym}$ -> $_{Target}$).Інша голова може фіксувати структурно-функціональні відносини (наприклад, $_{SMILE}$ -> $_{Target}$).Фінальний вихід механізму є конкатенацією результатів усіх голів, що забезпечує збагачене, багатомірне представлення для подальшої класифікації.

, (3.6)

де

Саме ця здатність до динамічного, багатоаспектного зважування та агрегації інформації від усіх чотирьох модальностей робить механізм Self-Attention незамінним інструментом для проєктування DDI-TransMDL.

Отже, Трансформер-Енкодер виступає оптимальним вибором для розв’язання задачі мультимодального прогнозування DDI, оскільки поєднує високу паралельність обчислень, глобальну рецептивність та здатність до динамічної інтеграції гетерогенних ознак. Це забезпечує формування глибоких, змістовних представлень, необхідних для точного та інтерпретованого багатоміткового прогнозування взаємодій лікарських засобів. Використання N=8 незалежних голів уваги дозволяє одночасно моделювати багатоаспектні відносини (наприклад, структурні, метаболічні та функціональні), створюючи таким чином збагачене, високоінформативне представлення . Саме ця здатність до глибокої та гнучкої міжмодальної інтеграції робить Трансформер-Енкодер оптимальним проєктним рішенням для багатоміткового прогнозування DDI.

3.2 Проєктування системи підготовки мультимодальних ознак

Проєктування моделі DDI-TransMDL вимагає створення надійної та структурованої системи для збору, вилучення та стандартизації гетерогенних даних. Ця система має забезпечити отримання чотирьох ключових модальностей (SMILE, Target, Enzyme, Pathway) та інформації про взаємодію з авторитетного фармакологічного джерела.

3.2.1 Проєктування механізму збору даних з DrugBank

Проєктування системи підготовки мультимодальних ознак для моделі DDI-TransMDL починається з розробки надійного механізму збору та вилучення даних. Як основне авторитетне джерело фармакологічної, хімічної та біологічної інформації, обрана база даних DrugBank. Проєктований механізм повинен забезпечити вилучення усіх чотирьох ключових модальностей (SMILE, Target, Enzyme, Pathway) та відповідних описів подій взаємодії.

Процес збору даних концептуально поділяється на три взаємопов'язані фази: ідентифікація цільової когорти ліків, вилучення індивідуальних мультимодальних ознак та вилучення структурованих даних про взаємодію.

Першочерговим завданням є формування вичерпного списку унікальних ідентифікаторів ліків (ID_{list}). Ця фаза передбачає ітеративний парсинг сторінок лістингу DrugBank, що охоплюють як Малі Молекулярні Ліки (Small Molecule Drugs), так і Біотехнологічні Ліки (Biotech Drugs). Проєктується використання HTTP-запитів до відповідних лістингових ендпойнтів із застосуванням пагінації для послідовного доступу до всіх сторінок. На кожній сторінці за допомогою методів веб-парсингу вилучаються унікальні ідентифікатори (Drug IDs) з гіперпосилань, що ведуть на індивідуальні профілі ліків.Для забезпечення стійкості до мережевих збоїв та обмежень сервера DrugBank, у проєкті закладається механізм ітеративного завантаження з повторними спробами (num_retries) та примусовими часовими затримками між запитами. Це є критично важливим для запобігання блокуванню та гарантування повноти зібраного списку ID_{list}.

На другому етапі для кожного (вилученні ознак), ідентифікатора з ID_{list} здійснюється доступ до індивідуальної сторінки профілю ліку з метою вилучення ознак, необхідних для моделі DDI-TransMDL. Проєктується вилучення таких основних компонентів:

- хімічні ознаки: лінійна SMILE-нотація, що відображає молекулярну структуру, вилучається з відповідного блоку ідентифікації.
- фармакологічні ознаки: ідентифікатори цільових білків (Targets), ферментів (Enzymes), а також переносників (Transporters) та носіїв (Carriers) вилучаються з блоку "Фармакологія". Усі вилучені ідентифікатори (наприклад, UniProt IDs) агрегуються у формати рядків-колекцій, де елементи розділені визначеним символом (наприклад, вертикальною рискою "|").

Вилучені атрибути (Drug ID, Name, SMILE, Target, Enzyme, Carrier, Transporter) негайно зберігаються у структурованій базі даних SQLite (таблиця drug). Це гарантує цілісність даних і надає платформу для подальшої обробки та генерації векторів ознак, як це описано в наступних підрозділах.

На етапі вилучення взаємодій для формування цільових міток Y та зв'язків між парами ліків проєктується механізм вилучення даних про взаємодію. Цей процес використовує спеціалізований API-ендпойнт DrugBank для отримання структурованих даних про DDI.Оскільки кількість взаємодій може бути значною, проєктом передбачено використання пагінації API з параметрами start та length для обробки великих наборів даних (наприклад, пачками по 100 записів). Кожен вилучений запис DDI надає інформацію про:

- ідентифікатор та назву ліку-ініціатора DA;
- ідентифікатор та назву ліку-субстрату DB4;
- повний текстовий опис події взаємодії(Interaction Description).

Вилучені записи зберігаються в окремій таблиці event бази даних SQLite, створюючи таким чином граф пар ліків та асоційованих текстових описів. Ці описи згодом будуть використані для мапінгу на C=65

уніфікованих класів подій DDI, що є основою для багатоміткової класифікації.

3.2.2 Формалізація мультимодальних вхідних даних

Після вилучення гетерогенних даних із DrugBank, наступний етап проєктування системи підготовки ознак полягає у формалізації цих даних у стандартизований числовий формат, придатний для подачі на вхід Трансформер-Енкодера. Ця формалізація має вирішити проблему різної природи даних (структурні, бінарні, набірні) та забезпечити єдину фіксовану розмірність для всіх модальностей.

Для модальностей, представлених як набори дискретних ідентифікаторів (Target, Enzyme, Pathway, а також хімічні фрагменти SMILE), проєктується механізм перетворення цих наборів у бінарні вектори фіксованої довжини. Це досягається через побудову словників ознак (Vocabularies), які відображають унікальні ідентифікатори (наприклад, DrugBank IDs ферментів) на індекси у векторі.

1. побудова словника: на основі всієї когорти ліків створюється словник $V_м$, що містить відображення Feature ID -> Index.
2. генерація вектора: вектор ознак для ліку D у модальності M генерується так, що елемент , якщо ознака, що відповідає індексу i, присутня у профілі ліку D і 0 в іншому випадку.

Оскільки початкова довжина векторів залежить від розміру словника || і може значно варіюватися між модальностями, висувається вимога до уніфікації розмірності. Це необхідно для забезпечення сумісності вхідних даних з фіксованою архітектурою Трансформера.

Проєктом встановлюється цільова розмірність ознак для одного ліку: = 572.

Для досягнення цієї розмірності застосовується така логіка:

- якщо || > (вектор надто великий), застосовується метод редукції розмірності або вибіркового ущільнення (в нашому випадку PCA) для зведення вектора до ;
- якщо || < (вектор занадто малий), застосовується доповнення нулями (Padding) до досягнення необхідної розмірності:

.

Вектор ознак для пари ліків у модальності M формується шляхом простої конкатенації стандартизованих векторів окремих ліків:

(3.7)

де позначає операцію конкатенації.

Це забезпечує, що кожен вхідний вектор для Трансформера 1144об'єднує повний профіль ознак обох ліків. Розмірність =1144 є фіксованою вхідною розмірністю для проєкційних шарів Трансформера.

Після конкатенації до фінального вектора додається явна ознака взаємодії – коефіцієнт Жаккарда , що вимірює спільність дискретних ознак між ліками. Проєктується інтеграція цієї скалярної метрики шляхом заміни останнього елемента конкатенованого вектора :

(3.8)

де - фінальний вектор ознак, готовий до подачі в трансформер. Така інтеграція гарантує, що модель отримує експліцитну інформацію про потенційну спільність механізмів дії ліків вже на вході.

Таким чином, для кожної пари генерується чотири фінальні вектори ознак { }, кожен розмірністю 1144.

3.2.3 Проєктування ознак взаємодії

Проєктування ефективної системи передбачення DDI вимагає не лише агрегації індивідуальних профілів ліків, але й створення механізму для явного кодування потенційної взаємодії між ними на рівні вхідних ознак. Це дозволяє моделі Трансформера з самого початку обробляти ознаки пари ліків як єдиний об'єкт із вбудованими метриками спільності.

Виходячи з вимоги уніфікації розмірності (Розділ 3.2.2), де кожен лік та має стандартизований вектор ознак для кожної модальності M, проєктується формування базового вектора пари через конкатенацію. Цей метод забезпечує пряме об'єднання профілів обох ліків в єдиний вхідний блок (Формула 3.7). У результаті, кожен базовий вектор пари має фіксовану розмірність . Ця структура
є необхідною для подачі на вхід проєкційним шарам Трансформера.

Ключовим проєктним рішенням є включення експліцитної ознаки взаємодії для підвищення чутливості моделі до спільності механізмів дії. Як ця ознака обрано Коефіцієнт Жаккарда . Коефіцієнт Жаккарда вимірює ступінь перекриття між наборами дискретних ознак (наприклад, Target IDs, Enzyme IDs, Pathway IDs) двох ліків. Його включення є обґрунтованим, оскільки висока спільність цілей або ферментів метаболізму є провідним фактором ризику DDI.

Формула розрахунку коефіцієнта Жаккарда для модальності M є класичною:

—, (3.9)

де |M(D)| – набір дискретних ознак ліку D у модальності M.

Проєктується інтеграція скалярного значення безпосередньо у вектор ознак . Замість збільшення загальної розмірності , що вимагало б змін у проєкційних шарах Трансформера, застосовується метод заміщення останнього елемента конкатенованого вектора.

Це створює фінальний вектор ознак розмірністю 1144, який має таку структуру:

, (3.10)

де останній елемент (який інакше був би просто елементом ознак) заміщується на .

Це проєктне рішення дає змогу:

1) зберегти уніфіковану вхідну розмірність =1144, відповідно до вимог архітектури трансформера;
2) гарантувати, що трансформер отримує експліцитну, нормалізовану метрику спільності, що спрощує для механізму уваги ідентифікацію потенційних конфліктів, спричинених спільними біологічними або хімічними ознаками.

Таким чином, для кожної з чотирьох модальностей генерується унікальний вектор , який є вхідним токеном для Трансформера.

3.2.4 Стратегія нормалізації та зменшення розмірності (Dimensionality Reduction)

Проєктування системи підготовки ознак передбачає не лише кодування та конкатенацію векторів, але й застосування стратегій для підвищення їхньої обчислювальної ефективності та якості. З цією метою висувається вимога щодо використання Аналізу Головних Компонент (PCA) та нормалізації.

Хоча механізм формалізації вхідних даних (Розділ 3.2.2) передбачає уніфікацію розмірності до =1144, початковий простір ознак, особливо після бінарного кодування рідкісних біологічних ідентифікаторів, може залишатися високорозмірним та розрідженим. Якщо розмірність вектора ознак пари значно перевищує 1144 (наприклад, при використанні надто великого словника SMILE-фрагментів або Pathway-ідентифікаторів), застосування PCA є обов'язковим.

Мета PCA у проєкті DDI-TransMDL полягає у такому.

1. Декореляція ознак**:** PCA трансформує вхідний простір, мінімізуючи кореляцію між вихідними ознаками. Це запобігає надмірному акценту на висококорельованих ознаках у процесі навчання трансформера, підвищуючи його здатність до узагальнення.
2. Збереження дисперсії: PCA гарантує, що при зменшенні розмірності (коли > 1144) зберігається максимальна інформаційна дисперсія в даних, усуваючи при цьому шум, що міститься у малоінформативних, низькодисперсійних компонентах.

3. Фіксація розмірності: Трансформер-Енкодер вимагає фіксованої вхідної розмірності. PCA слугує механізмом для проєкції будь-якого вектора у цільовий простір :

, (3.11)

де – матриця головних компонент, навчена на тренувальному наборі даних.

Незалежно від застосування PCA, для забезпечення стабільності градієнтів під час навчання Трансформера, проєктується обов'язкове використання масштабування (Normalization) вхідних ознак.

1. Стандартизація: Вектори ознак (особливо після PCA, або якщо вони містять чисельні значення, такі як Жаккард) повинні бути стандартизовані. Це забезпечує нульове середнє значення (= 0) та одиничну дисперсію (для кожного елемента вектора ознак.

–, (3.12)

2. Обґрунтування: Трансформер-Енкодер використовує механізми Layer Normalization у своїх шарах. Однак, попереднє масштабування вхідних даних (Input Normalization) є критично важливим для прискорення збіжності та забезпечення того, що початкові ваги проєкційних шарів рівномірно впливають на всі модальності.

Таким чином, фінальний вектор , який подається на вхід Трансформера, є стандартизованим, а його розмірність гарантовано оптимізована з точки зору інформаційної щільності та обчислювальної ефективності.

3.3 Архітектура трансформерного енкодера DDI-TransMDL

Проєктування вхідного шару Трансформера є критично важливим для перетворення високорозмірних, гетерогенних векторів ознак у єдиний, низькорозмірний простір, де механізм уваги може ефективно працювати.

Для кожної з чотирьох стандартизованих вхідних модальностей ( ), де вектор ознак пари має фіксовану розмірність = 1144, проєктується окремий лінійний проєкційний шар.

Кожен такий шар, , виконує операцію матричного множення:

, (3.13)

де – вхідний вектор ознак пари для модальності M, – навчена вагома матриця, унікальна для модальності M, – вектор зміщення (bias), – вектор вбудовування (embedding) модальності M, що має цільову розмірність = 256.

Таким чином, проєктується чотири окремі, незалежні лінійні шари: .

Використання індивідуальних проєкційних шарів для кожної модальності, замість одного спільного шару, є свідомим проєктним рішенням, обґрунтованим гетерогенністю вхідних даних.

1. Ознаки різних модальностей несуть якісно різну інформацію. Наприклад, бінарний вектор описує метаболічну активність, тоді як відображає хімічні фрагменти. Ці ознаки оперують у різних семантичних просторах.
2. Унікальна трансформація: індивідуальна вагома матриця дозволяє моделі вивчити унікальну трансформацію, необхідну для "перекладу" семантики конкретної модальності (наприклад,

Enzyme) у спільний мовний простір . Це гарантує, що втрата інформації при зменшенні розмірності (з 1144 до 256) буде мінімальною і специфічною для типу ознак.

3. Застосування окремих шарів розділяє складне завдання відображення гетерогенних даних на чотири простіші підзадачі. Це підвищує стабільність градієнтів та ефективність навчання на ранніх етапах.

Таким чином, індивідуальна проєкція забезпечує, що перед подачею в механізм Self-Attention, кожен токен є високоякісним, оптимізованим представленням своєї модальності, готовим до міжмодальної взаємодії.

3.3.2 Токен класифікації ([CLS] Token) та позиційне кодування

Після перетворення індивідуальних векторів ознак кожної модальності у простір вбудовування , необхідно підготувати фінальну послідовність, що буде подана на вхід Трансформер-Енкодеру. Цей етап проєктування включає інтеграцію спеціального токена класифікації та додавання інформації про позицію модальностей.

Проєктом передбачається включення спеціального навчального токена [CLS] (Classification Token) на початок вхідної послідовності. Його основна функціональна роль полягає у агрегації контекстуальної інформації від усіх інших модальностей протягом проходження через шари Трансформера.

Токен [CLS] ініціалізується як навчальний вектор . Його початкове значення є довільним і не несе семантичного навантаження.

У кожному шарі трансформера, токен [CLS] активно взаємодіє з іншими токенами модальностей ( ) через

механізм Self-Attention. Він використовує свій вектор Запиту ( ) для збору інформації з векторів Значення (V) усіх інших модальностей.

Після проходження шарів Енкодера, вихідний вектор токена [CLS] ( ) інкапсулює інтегроване, багатоаспектне представлення пари ліків, що враховує взаємодію всіх чотирьох модальностей.

Таким чином, стає єдиним, фіксованим вектором представлення пари ліків, який подається на вхід Класифікаційній Голові для прогнозування C=65 DDI-подій.

Фінальна вхідна послідовність для Трансформера формується шляхом конкатенації токена [CLS] та вбудовувань модальностей:

.

Оскільки Трансформер-Енкодер не містить рекурентних або згорткових шарів, він не має вбудованого механізму для врахування порядку або позиції токенів у послідовності. У проєкті DDI-TransMDL порядок модальностей не є часовим, але є семантичним (наприклад, [CLS] завжди стоїть на першій позиції).

Проєктується додавання навчального позиційного кодування до вхідної послідовності. Це є набір навчальних векторів, кожен з яких відповідає певній позиції (типу модальності) у послідовності:

, (3.14)

де : відповідає токену [CLS], відповідає токену SMILE, відповідає токену Target і т.д.

Додавання дозволяє моделі розрізняти модальності та інформує механізм уваги про те, який тип ознак (наприклад, Enzyme чи Target)

знаходиться на конкретній позиції. Це посилює здатність Self-Attention до правильного зважування міжмодальних взаємозв'язків.

3.3.3 Проєктування шару Multi-Head Self-Attention (MHSA)

Механізм Multi-Head Self-Attention (MHSA) є ядром архітектури Трансформер-Енкодера і ключовим елементом, що забезпечує глибоку міжмодальну інтеграцію в моделі DDI-TransMDL. Проєктується використання ідентичних шарів Енкодера, кожен з яких містить блок MHSA, який працює з вектором розмірності =256.

Основний принцип механізму Self-Attention полягає у зваженій агрегації інформації. Для кожного токена (модальності) у послідовності , механізм обчислює, наскільки сильно він повинен "приділяти увагу" всім іншим токенам , включаючи [CLS] та самого себе.

Цей процес вимагає проєкції кожного вхідного вектора у три різних простори: Запит (Q, Query), Ключ (K, Key) та Значення (V, Value):

, (3.15)

де – навчені вагові матриці.

Далі обчислюються коефіцієнти уваги A, які визначають матрицю подібності між усіма парами токенів, та фінальний вихід :

, (3.16)

, (3.17)

де , а L = 5 – довжина послідовності, – фактор масштабування, що запобігає перенасиченню функції softmax.

Для підвищення здатності моделі до виявлення різноманітних міжмодальних залежностей проєктується механізм Multi-Head Attention з = 8 незалежними головами.

Загальна розмірність моделі рівномірно розподіляється між вісьмома головами, де розмірність кожної голови становить: .

Кожна голова обчислює свій власний незалежний набір проєкцій ( ) та свій вихід уваги:

, (3.18)

де є унікальними ваговими матрицями для кожної голови.

Вихід механізму MHSA є результатом конкатенації виходів усіх голів, що потім проєктується назад у простір за допомогою фінальної вагомої матриці :

. (3.19)

Цей механізм гарантує, що модель одночасно фокусується на восьми різних аспектах взаємодії модальностей.

Після MHSA кожен шар Енкодера охоплює:

- застосовується механізм Add & Norm, що забезпечує стабільність навчання та полегшує поширення градієнтів через глибоку мережу;
- проєктується позиційно-незалежна FFN, що застосовується до кожного токена; Вона складається з двох лінійних шарів з проміжним підвищенням розмірності та функцією активації

GELU (Gaussian Error Linear Unit) для введення нелінійності. FFN дозволяє моделі обробляти внутрішні, незалежні перетворення кожного токена.

3.3.4 Проєктування шару трансформер-енкодера

Проєктування повного шару трансформер-енкодера (Transformer Encoder Layer) DDI-TransMDL є об'єднанням трьох основних компонентів: механізму Multi-Head Self-Attention (MHSA), шару прямого зв'язку (Feed-Forward Network, FFN) та механізмів нормалізації і залишкових зв'язків (Add & Norm). Проєктується використання =3 ідентичних шарів, які працюють з розмірністю вбудовування .

Кожен шар трансформер-енкодера складається з двох основних підшарів, кожен з яких супроводжується залишковим зв'язком (Residual Connection) та нормалізацією шару (Layer Normalization).

, (3.20)

де – вихід попереднього шару, а – вихід поточного шару .

Перший підшар є вже детально описаним механізмом MHSA (пункт 3.3.3). Його функція – обчислення динамічних зважених сум, що інтегрують інформацію від усіх п'яти токенів послідовності (CLS та чотирьох модальностей). Вихід MHSA додається до входу цього підшару (або виходу попереднього шару). Застосовується Layer Normalization для стандартизації активацій.

Другий підшар – це позиційно-незалежна FFN, яка застосовується до кожного токена (вектора) послідовності незалежно. FFN є двошаровою

мережею, що забезпечує нелінійну трансформацію представлень токенів. Вектор (розмірність) спочатку проєціюється в проміжний простір, зазвичай у 4 рази більший за (тобто 256 x 4 = 1024). Це збільшення внутрішньої розмірності дозволяє моделі вивчати складніші нелінійні взаємозв'язки. Між лінійними шарами застосовується функція GELU (Gaussian Error Linear Unit), яка є вдосконаленням ReLU, що забезпечує плавнішу нелінійність. На фінальному лінійному шарі розмірність повертається до .

. (3.21)

Вихід FFN додається до входу цього підшару ( ), і застосовується фінальна нормалізація.

. (3.21)

Після проходження через всі три шари Енкодера, перший вектор послідовності буде містити фінальне, висококонтекстуалізоване представлення пари ліків, готове для подачі на класифікаційну голову.

3.4 Проєктування класифікаційної голови та функції втрат

3.4.1 Механізм класифікації

Після того, як мультимодальні ознаки пари ліків були інтегровані та трансформовані через 3 шари Трансформер-Енкодера, необхідно перетворити агреговане представлення у фінальний прогноз щодо 65 DDI-подій. Це завдання виконує Класифікаційна Голова (Classification Head).

Механізм класифікації проєктується на основі фінального вектора токена [CLS]. Цей вектор, отриманий на виході останнього шару Енкодера, інкапсулює зважену та контекстуалізовану інформацію про взаємозв'язки між усіма чотирма модальностями для даної пари ліків. Його використання є стандартним підходом у трансформер-архітектурах для задач класифікації.

,

де є першим елементом (токен [CLS]) у вихідній послідовності Енкодера.

Класифікаційна голова проєктується як двошарова нелінійна мережа, що має на меті зменшити розмірність (з) та відобразити його у простір C=65 DDI-подій.

Застосовується лінійне перетворення для зменшення розмірності та введення нелінійності.

- вхідна розмірність: ;
- вихідна розмірність: ;
- активація: використовується функція GeLU для нелінійності;
- регуляризація: застосовується Dropout для запобігання перенавчанню.

(3.22)

Фінальний шар проєктує проміжний вихід у простір подій DDI.

- вхідна розмірність: 128;
- вихідна розмірність: 65 (кількість DDI-подій);

Оскільки DDI-прогноз є задачею багатоміткової бінарної класифікації (одна пара ліків може спричинити декілька подій одночасно), на фінальному виході застосовується функція Sigmoid до кожного з 65 виходів незалежно.

, (3.23)

де є вектором прогнозованих ймовірностей. Кожен елемент є ймовірністю настання конкретної DDI-події c.

3.4.2 Визначення кількості виходів та типу задачі

Цей підрозділ формалізує кінцеву мету моделі DDI-TransMDL, визначаючи точний характер класифікаційної задачі та обґрунтовуючи кількість необхідних виходів.

Прогнозування взаємодії ліків є задачею багатоміткової бінарної класифікації (Multi-label Binary Classification). Це принципово відрізняється від традиційної багатокласової класифікації (Multi-class Classification), де об'єкт може належати лише до одного класу. У контексті DDI, пара ліків (D_A, D_B) може одночасно спричиняти декілька клінічних подій. Наприклад, взаємодія може призвести як до "зниження ефективності антикоагулянта", так і до "підвищення ризику кровотечі".

Вимога до моделі полягає у незалежному прогнозуванні ймовірності для кожного можливого класу подій.

Кількість виходів C безпосередньо відповідає кількості унікальних DDI-подій, які було ідентифіковано та уніфіковано з вихідних текстових описів бази даних DrugBank.

Проєктом встановлено, що після кластеризації та уніфікації текстових описів взаємодії, система повинна прогнозувати 65 унікальних класів подій. Це фіксоване число визначає необхідну розмірність вихідного шару класифікаційної голови

Для забезпечення незалежного прогнозування ймовірності для кожного з 65 виходів, на фінальному лінійному шарі застосовується функція активації Sigmoid:

, ——— (3.24)

де – логіт (вихід лінійного шару) для *c*-ї події, – прогнозована ймовірність.

Вибір Sigmoid є необхідним, оскільки, на відміну від функції softmax (яка використовується у багатокласовій класифікації і гарантує, що , Sigmoid дозволяє сумі ймовірностей перевищувати одиницю, що відповідає вимозі про незалежність подій.

3.4.3 Проєктування функції втрат

Проєктування функції втрат (Loss Function) є критичним етапом, оскільки вона керує процесом навчання, мінімізуючи розбіжності між прогнозованими ймовірностями та істинними бінарними мітками . Вибір функції втрат безпосередньо залежить від формату задачі, визначеної у пункті 3.4.2.

Для багатоміткової класифікації, де кожен із 65 виходів є незалежним бінарним рішенням, проєктом обирається функція Binary Cross-Entropy Loss ( ). Цей вибір є необхідним, оскільки він розглядає кожен клас (DDI-подію) як незалежну задачу бінарної класифікації.

Функція обчислюється як сума втрат за кожен клас j та кожен приклад і у батчі:

— , (3.25)

де N – розмір батчу (кількість пар ліків), C=65 – загальна кількість DDI-подій, – істинна бінарна мітка для i-го прикладу та j-ї події, – прогнозована ймовірність для i-го прикладу та j-ї події, отримана після активації Sigmoid.

Оскільки застосовується індивідуально до кожного класу, вона ідеально відповідає вимозі про незалежність DDI-подій. Якщо пара ліків має мітку '1' для події A та '0' для події B, втрати за подіями A і B обчислюються окремо, і градієнти, що йдуть назад від цих втрат, не впливають на відносні ймовірності інших класів.

Функція ефективно карає модель за дві основні помилки:

- коли істинна мітка , але модель прогнозує (висока втрата).
- коли істинна мітка , але модель прогнозує (висока втрата).

Проєктується потенційна модифікація для обліку дисбалансу класів. Оскільки більшість DDI-подій є рідкісними, частота появи міток '1' для деяких із 65 класів може бути вкрай низькою.

Для запобігання домінуванню градієнтів від частих негативних прикладів ( ), проєктується можливість використання збалансованої BCELoss, де вводяться вагові коефіцієнти :

— , (3.26)

де – ваговий коефіцієнт, обернено пропорційний частоті появи події j у тренувальному наборі. Це забезпечує більший акцент на правильному прогнозуванні рідкісних, але клінічно важливих DDI-подій.3

3.5 Проєктування протоколу навчання та оцінювання

3.5.1 Стратегія валідації

Для забезпечення надійної та неупередженої оцінки узагальнюючої здатності моделі DDI-TransMDL проєктується сувора стратегія валідації, заснована на K-Fold Cross-Validation з механізмом Early Stopping.

Використання K-Fold Cross-Validation з K=5 (п'ять фолдів) є обов'язковим для оцінки ефективності моделі на обмеженому, але цінному, фармакологічному наборі даних. Цей підхід гарантує, що кожна пара ліків із набору даних буде використана:

1) точно один раз як тестовий приклад (для фінальної оцінки у фолді);
2) K-1 разів як тренувальний або валідаційний приклад.

Це мінімізує зміщення оцінки (Bias) порівняно з простим випадковим поділом на єдиний тренувальний/тестовий набір.

Проєктна схема поділу даних на k-му фолді:

- : використовується для оновлення ваг моделі;
- : використовується для моніторингу прогресу навчання та механізму Early Stopping; це піднабір (наприклад, 20% від тренувальних даних);
- : використовується лише один раз наприкінці навчання фолду для отримання неупередженої оцінки метрик.

Для запобігання перенавчанню (Overfitting) моделі на тренувальних даних проєктується механізм Early Stopping, що контролюється на валідаційному наборі.

Як основна метрика для моніторингу обирається F1-Score (Micro-Averaged), оскільки вона є збалансованою мірою точності (Precision) та повноти (Recall) і більш надійно відображає якість багатоміткової класифікації, ніж проста точність (Accuracy). Моніторинг здійснюється протягом кожної епохи. Навчання припиняється, якщо валідаційний F1-Score не покращується протягом заданого числа епох (наприклад, Patience = 5). На кожній епосі, де валідаційний F1-Score досягає нового максимуму ( ), зберігаються поточні вагові коефіцієнти моделі. Наприкінці навчання (після спрацьовування Early Stopping), для фінальної оцінки на тестовому наборі використовується саме ця найкраща збережена версія моделі.

Фінальна продуктивність моделі DDI-TransMDL визначається як усереднені значення усіх ключових метрик (ROC-AUC, AUPR, F1-Score) по всіх K=5 тестових наборах.

– (3.27)

3.5.2 Вибір оптимізатора та планувальника швидкості навчання

Ефективність навчання моделі DDI-TransMDL Трансформер-Енкодера значною мірою залежить від правильного вибору алгоритму оптимізації та стратегії управління швидкістю навчання. Ці компоненти критично важливі для забезпечення швидкої збіжності та запобігання застряганню у локальних мінімумах складного простору втрат.

Як основний алгоритм оптимізації проєктується Adam (Adaptive Moment Estimation). Adam є адаптивним методом оптимізації, який обчислює індивідуальні швидкості навчання для різних параметрів мережі, ґрунтуючись на оцінках перших моментів (середнього) та других моментів

(дисперсії) градієнтів. Це особливо важливо для Трансформерів, оскільки різні шари (наприклад, MHSA та FFN) і навіть різні компоненти в межах одного шару (Query, Key, Value) вимагають різних темпів оновлення.

Для боротьби з перенавчанням та контролю складності моделі Adam використовується з інтегрованою L2-регуляризацією (Weight Decay). Вона додає до функції втрат штраф, пропорційний квадрату норми ваг , що сприяє отриманню менших за величиною ваг і підвищує узагальнюючу здатність.

, (3.28)

де – коефіцієнт згасання.

Стратегія управління швидкістю навчання ( ) проєктується як двокомпонентний підхід, спрямований на стабілізацію навчання на ранніх етапах та забезпечення ефективного пошуку мінімуму на пізніх. На початку навчання (перші епох) швидкість навчання повинна поступово лінійно зростати від нуля до початкового цільового значення .

, для ,

де – поточна епоха. Обґрунтуванням є необхідність запобігти великим стрибкам градієнтів, що можуть дестабілізувати мережу з випадково ініціалізованими вагами.

Після завершення фази Warmup, застосовується адаптивний планувальник ReduceLROnPlateau. Цей механізм контролює валідаційну метрику (F1-Score). Якщо метрика не покращується протягом заданого періоду очікування (Patience), швидкість навчання автоматично зменшується на фіксований коефіцієнт (наприклад, на 0.5). Це дозволяє моделі вийти з

плато, використовуючи менші кроки для точного наближення до мінімуму функції втрат.

Оскільки Трансформер-архітектури, особливо з механізмом Self-Attention, можуть генерувати надзвичайно великі градієнти (вибухові градієнти), проєктується обов'язкове використання обрізання градієнтів (Gradient Clipping). Обрізання гарантує, що норма градієнтів не перевищує встановлений максимальний поріг .

3.5.3 Механізм запобігання перенавчанню

Проєктування моделі DDI-TransMDL повинно включати комплексні механізми для запобігання перенавчанню (Overfitting), що є поширеною проблемою у глибоких мережах, особливо при роботі з обмеженими за розміром біологічними наборами даних. Ці механізми спрямовані на підвищення узагальнюючої здатності моделі.

Основний механізм регуляризації, закладений у проєкт, - це Dropout. Він застосовується як у шарах Трансформер-Енкодера, так і в Класифікаційній Голові.

Під час фази навчання Dropout випадковим чином "вимкне" (обнулить) вихідні нейрони з певною ймовірністю .Це означає, що певна частка нейронів тимчасово виключається з мережі. Dropout змушує нейронну мережу уникати надмірної залежності від окремих ознак або від коаліції специфічних нейронів. Завдяки цьому, модель вивчає більш надійні та стійкі представлення ознак, підвищуючи свою стійкість до варіацій у вхідних даних.

Як було визначено у пункті 3.5.2, L2-регуляризація (Weight Decay) інтегрована в оптимізатор Adam. Цей механізм додає до функції втрат штраф, пропорційний квадрату норми ваг (формула 3.28).

– коефіцієнт регуляризації. L2-регуляризація примушує модель надавати перевагу меншим за величиною ваговим коефіцієнтам. Це ефективно зменшує складність моделі, згладжує простір рішень та запобігає ситуації, коли модель може використовувати надзвичайно великі ваги для точного запам'ятовування тренувальних прикладів.

Третій, і найважливіший, механізм запобігання перенавчанню - це Early Stopping (Рання Зупинка), детально описаний у підрозділі 3.5.1. Механізм моніторить продуктивність моделі на валідаційному наборі протягом кожної епохи. Навчання автоматично припиняється, як тільки валідаційна втрата починає зростати або цільова метрика (F1-Score) припиняє покращуватися протягом заданого періоду очікування (Patience). Це зупиняє процес навчання саме в момент, коли модель починає "запам'ятовувати" шум тренувальних даних, тим самим запобігаючи перенавчанню.

Комбінація внутрішньої стохастичної регуляризації (Dropout), штрафування складності ваг (L2-Regularization) та зовнішнього контролю збіжності (Early Stopping) забезпечує надійний захист моделі DDI-TransMDL від перенавчання.

3.5.4 Проєктування оціночних метрик

Проєктування оціночних метрик є ключовим для об'єктивного вимірювання продуктивності моделі DDI-TransMDL. Оскільки задача є багатомітковою бінарною класифікацією (65 незалежних подій), вибір метрик має відображати як загальну ефективність моделі, так і її здатність до прогнозування рідкісних, але клінічно значущих подій.

Проєктом передбачається використання двох основних груп метрик: порогово-залежні (на основі бінарних прогнозів) та порогово-незалежні (на основі ймовірностей).

Ці метрики вимагають перетворення прогнозованих ймовірностей у бінарні мітки за допомогою фіксованого порогу, наприклад, 0.5. Precision - частка правильно передбачених позитивних міток серед усіх міток, передбачених моделлю як позитивні. Recall - частка правильно передбачених позитивних міток серед усіх істинно позитивних міток. F1-Score - Гармонійне середнє Precision та Recall, що є збалансованим показником якості бінарної класифікації.

Для багатоміткової задачі обов'язково проєктується розрахунок метрик у двох режимах усереднення.

Micro-усереднення: обчислюється шляхом зведення показників TP (True Positives), FP (False Positives) та FN (False Negatives) по всіх C класах:

. (3.29)

Micro-F1 є основною метрикою, оскільки вона відображає загальну продуктивність моделі та ефективність на великих, частих класах.

Macro-усереднення: Обчислюється шляхом знаходження метрики для кожного класу окремо, а потім усереднення результатів.

– (3.30)

Macro-F1 є важливим для оцінки здатності моделі до прогнозування рідкісних DDI-подій, оскільки він надає однакову вагу як великим, так і малим класам.

AUC та AUPR метрики використовують безпосередньо прогнозовані ймовірності , що забезпечує більш повну картину якості ранжування моделі.

Проєктом вимагається фінальне усереднення всіх метрик, отриманих на тестових наборах з усіх K=5 фолдів. У звіті мають бути представлені:

- Micro Precision, Recall, F1-Score;
- Micro ROC-AUC, Micro AUPR (як основні показники загальної ефективності);
- Macro F1-Score, Macro AUPR (як показники якості на рідкісних клінічних подіях).

Висновки до розділу 3

Розділ 3 детально окреслив архітектурне та методологічне проєктування моделі DDI-TransMDL, що є багатообіцяючим підходом до вирішення складної задачі багатоміткового прогнозування взаємодії лікарських засобів. Ключові проєктні рішення, починаючи від збору даних і закінчуючи оціночними метриками, були спрямовані на створення робастної, інтерпретованої та високопродуктивної моделі.

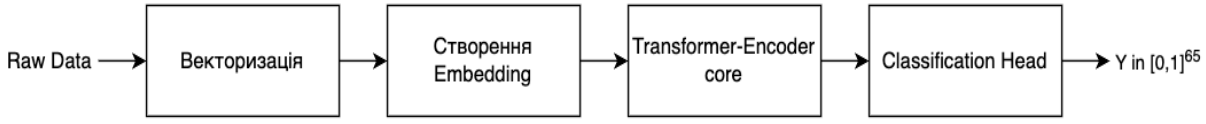

Рисунок 3.1 – Загальна топологія моделі

Замість традиційних одномодальних підходів, проєкт впроваджує інтеграцію чотирьох гетерогенних модальностей (SMILE, Target, Enzyme, Pathway). Це відображає комплексний характер DDI, де різні аспекти біологічної активності та хімічної структури ліків взаємодіють.

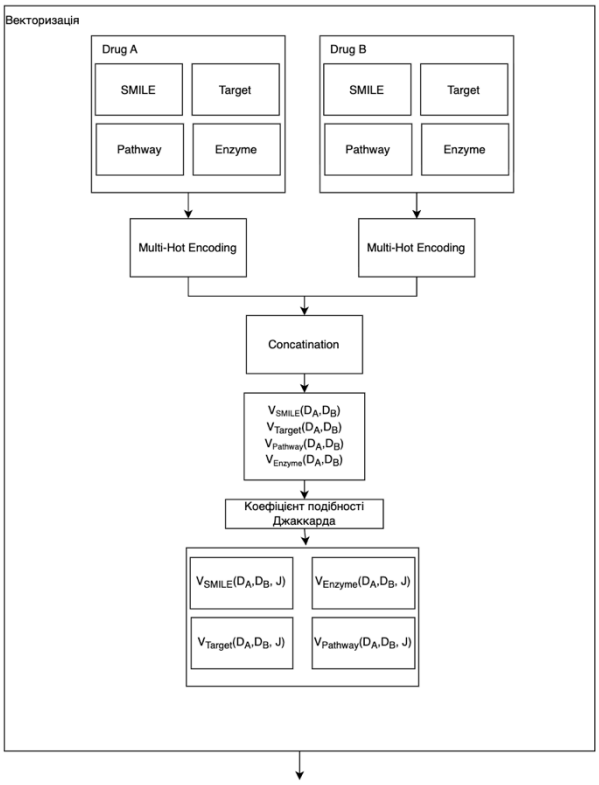

Рисунок 3.2 – Векторизація вхідних даних

Вибір архітектури Трансформер-Енкодера з механізмом Multi-Head Self-Attention є фундаментальним. Це дозволяє моделі не просто об'єднувати ознаки, а динамічно зважувати та контекстуалізувати взаємозв'язки між різними модальностями, забезпечуючи глибоке вивчення нелінійних залежностей. Роль токена [CLS] як агрегатора інформації підкреслює ефективність цього підходу.

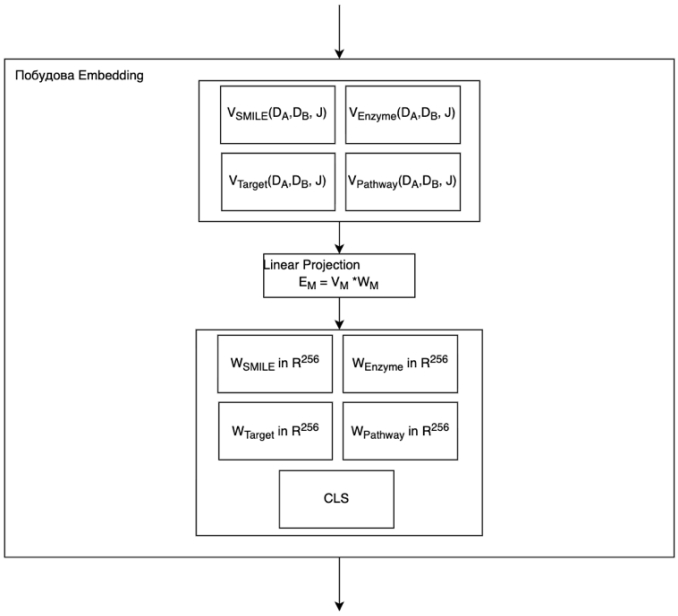

Рисунок 3.3 – Побудова Embedding векторів

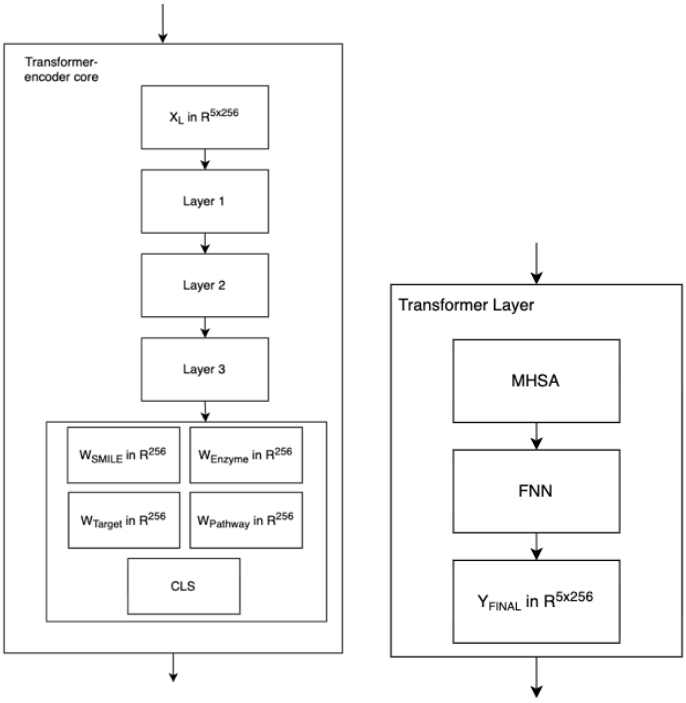

Рисунок 3.4 – Архітектура трансформера (ліворуч) та архітектура шару трансформеру (праворуч)

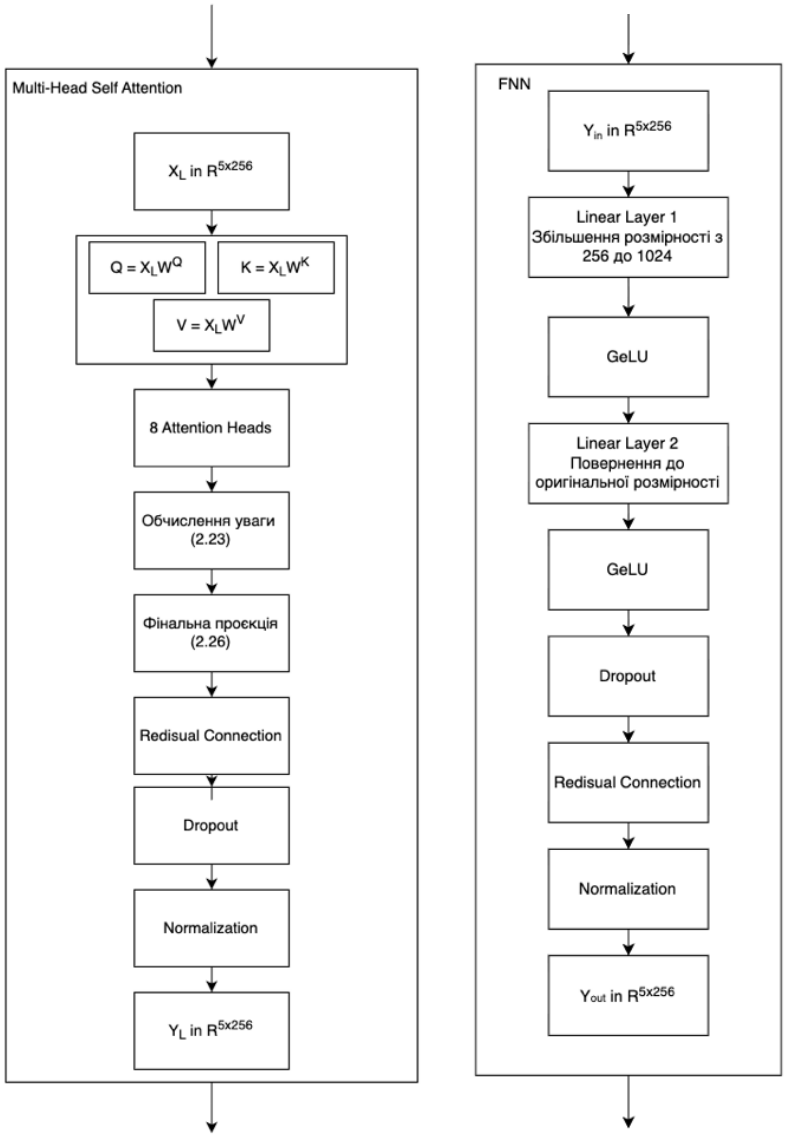

Рисунок 3.5 – Топологія блоків MHSA (ліворуч) та FNN (праворуч)

Інтеграція коефіцієнта Жаккарда як експліцитної ознаки подібності між ліками для кожної модальності підвищує здатність моделі виявляти ризики DDI на ранніх етапах.

Формалізація задачі як багатоміткової бінарної класифікації з C=65 незалежними DDI-подіями дозволяє моделі надавати вичерпний прогноз, що має високу клінічну цінність.

Проєктування механізмів збору даних з DrugBank, стратегій нормалізації та зменшення розмірності (PCA), а також суворої стратегії K-

Fold Cross-Validation з Early Stopping та комплексних оціночних метрик (Micro/Macro F1-Score, ROC-AUC, AUPR) гарантує надійність моделі та об'єктивність її оцінки.

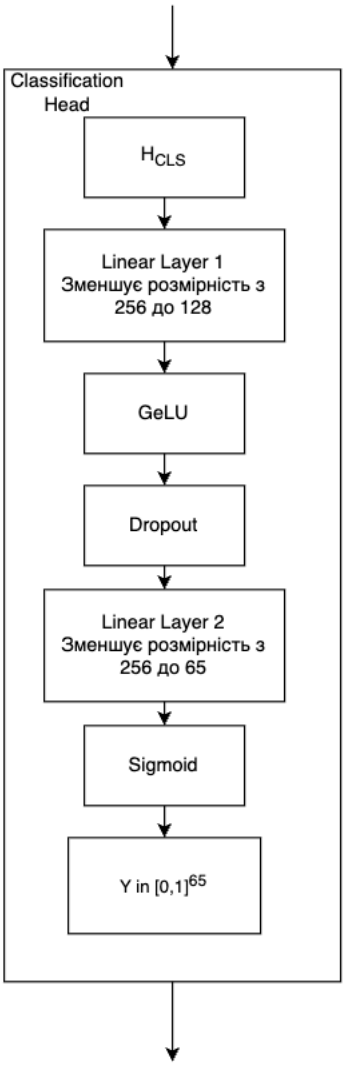

Рисунок 3.8 – Архітектура Classification Head

Таким чином, DDI-TransMDL є архітектурним рішенням, яке поєднує передові методи глибокого навчання з ретельним врахуванням специфіки фармакологічних даних. Його проєктування спрямоване на створення моделі, яка не тільки досягне високої прогностичної точності, але й буде достатньо гнучкою для адаптації до різноманітних клінічних сценаріїв DDI.

РОЗДІЛ 4 ІМПЛЕМЕНТАЦІЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ МОДЕЛІ DDI-TRANSMDL

Технічна імплементація моделі DDI-TransMDL ґрунтується на архітектурних вимогах, детально описаних у Розділі 3, і виконується з використанням спеціалізованих бібліотек глибокого навчання.

4.1 Технічна імплементація моделі

Вся імплементація виконана мовою Python, завдяки її гнучкості, широкій підтримці спільноти та багатому набору бібліотек для наукових обчислень та машинного навчання.

Як основний фреймворк для побудови та навчання нейронної мережі обрано PyTorch. Його переваги включають: динамічні обчислювальні графи та обширна підтримка GPU.

Проєктна вимога щодо ефективності навчання виконана шляхом використання апаратного прискорення. Вся модель та тензори даних (вектори ознак та мітки) переносяться на графічний процесор (GPU) за допомогою об'єкта *DEVICE = torch.device('cuda' if torch.cuda.is_available() else 'cpu')*. Це є необхідним для скорочення часу навчання глибокої мережі, яка оперує векторами розмірністю 1144. Для ефективної підготовки даних, обробки великих масивів та реалізації допоміжних функцій використовуються:

- NumPy: для базових числових операцій, маніпуляції векторами ознак та розрахунку метрик;

- Pandas: для завантаження, структурування та попередньої обробки даних з бази SQLite;
- Scikit-learn (Sklearn): використовується для реалізації процедур крос-валідації (KFold) та обчислення фінальних метрик оцінки (ROC-AUC, AUPR).

Для зберігання витягнутих мультимодальних ознак та структурованих описів взаємодії (DDI events) використовується база даних SQLite.

Для моніторингу процесу навчання та формування фінальних звітів про продуктивність використовуються Matplotlib та Seaborn. Вони забезпечують генерацію діаграм прогресу навчання (Validation Loss, F1-Score) та візуалізацію фінальних результатів (Box plots, Bar charts).

4.2 Результати крос-валідації та оцінка ефективності

4.2.1 Порівняльний аналіз метрик по фолдах

Порівняльний аналіз результатів, отриманих на п'яти незалежних тестових наборах крос-валідації (K=5), є критично важливим для підтвердження стійкості та надійності моделі DDI-TransMDL. Висока однорідність метрик між фолдами свідчить про те, що модель не перенавчається на специфічних підмножинах даних і демонструє гарну узагальнюючу здатність.

Значення ROC-AUC є надзвичайно високими та стабільними, коливаючись у вузькому діапазоні від 0.988 до 0.990. Це свідчить про те, що модель послідовно досягає майже ідеального ранжування позитивних пар ліків вище, ніж негативних, незалежно від того, який фолд використовувався для тестування.

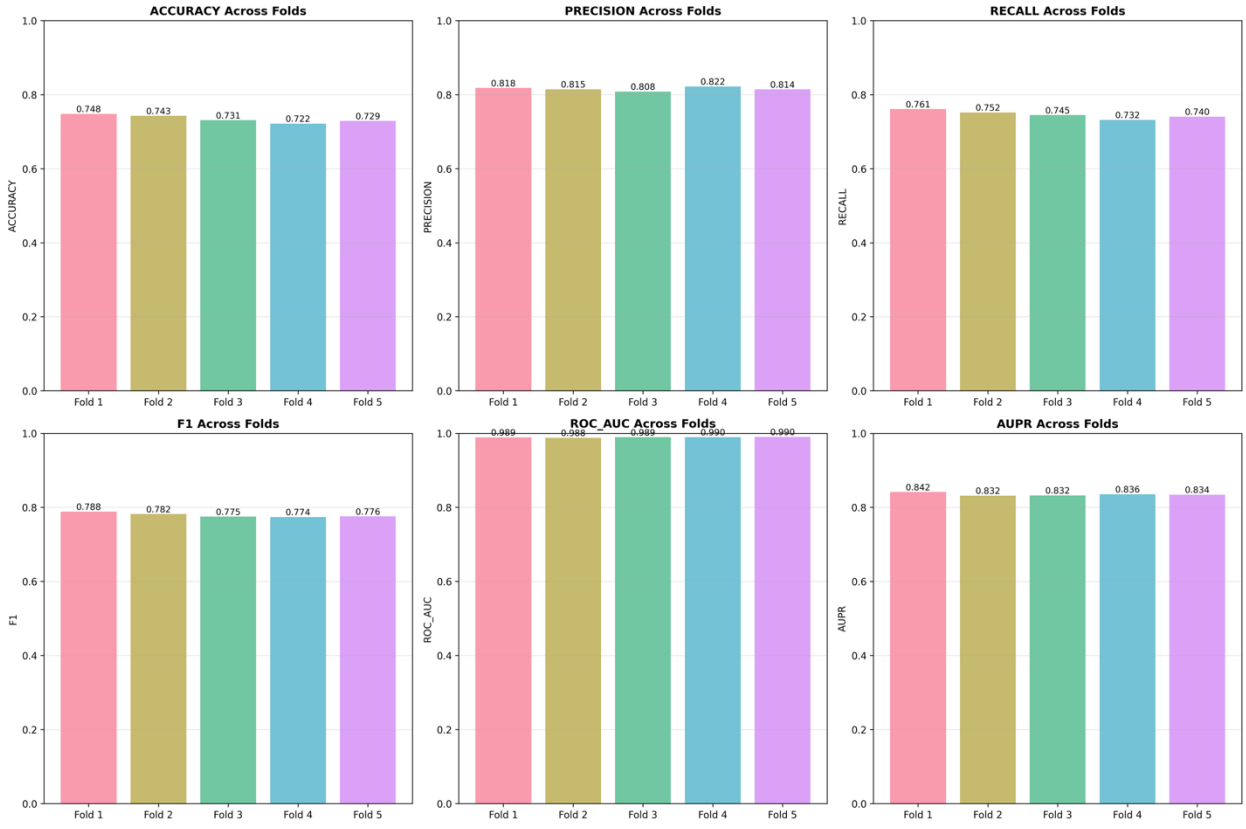

Рисунок 4.1 – Гістограми пофолдових метрик

Метрика AUPR також демонструє високу стійкість, коливаючись від 0.832 до 0.842. Це підтверджує, що модель ефективно працює на умовах незбалансованих класів, що є типовим для DDI-прогнозування.

F1-Score коливається від 0.774 до 0.788, а Accuracy - від 0.722 до 0.748. Невелика варіативність у цих метриках є нормальною і свідчить про те, що модель послідовно зберігає баланс між Precision та Recall.

Для ROC-AUC, Precision та AUPR діапазон між першим та третім квартилями (IQR) є дуже вузьким. Розподіл практично зливається у вузькій смузі, близькій до 1.00 (середнє значення ≈0.9893), що вказує на виняткову стійкість моделі щодо цієї метрики.

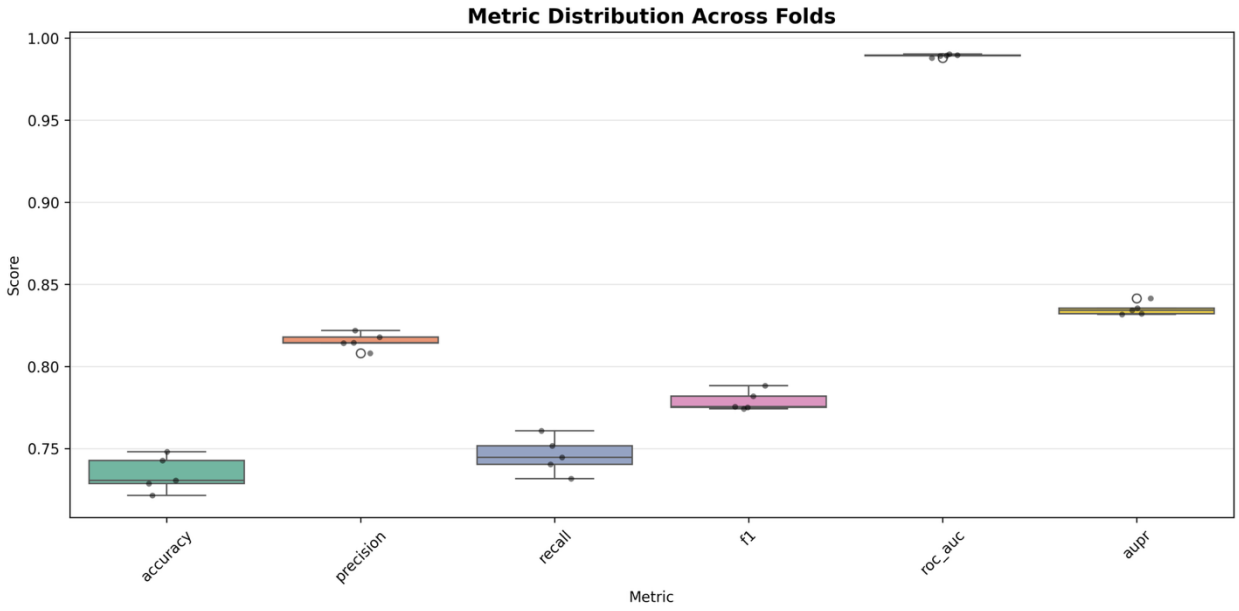

Рисунок 4.2 – Box Plot фолдів

Ці метрики також мають компактний розподіл (Precision ≈ 0.815, AUPR ≈ 0.8351), що свідчить про низьку дисперсію результатів крос-валідації. Медіанні лінії (внутрішні лінії ящиків) для всіх метрик розташовані близько до середнього значення, підтверджуючи, що результати кожного фолду є репрезентативними.Наявність мінімальної кількості викидів (кіл) підтверджує, що жоден із п'яти фолдів не дав аномально поганого або надмірно гарного результату, що зміщував би фінальну усереднену оцінку.

Однорідність результатів між фолдами, підтверджена вузьким діапазоном і компактним розподілом на Box Plots, свідчить про те, що модель DDI-TransMDL має високу узагальнюючу здатність. Це означає, що механізм Multi-Head Self-Attention успішно вивчив фундаментальні, а не специфічні для тренувального набору, взаємозв'язки між мультимодальними ознаками.

4.2.2. Фінальні Усереднені Результати

Фінальні усереднені результати, отримані шляхом агрегації метрик з п'яти незалежних тестових фолдів, є основним показником загальної ефективності та узагальнюючої здатності моделі DDI-TransMDL. Аналіз цих результатів проводиться у двох ключових режимах: Micro-усереднення (загальна ефективність) та Macro-усереднення (ефективність на рідкісних класах).

Micro-усереднені метрики розраховуються шляхом об'єднання показників True Positives (TP), False Positives (FP) та False Negatives (FN) по всіх 65 класах. Вони відображають загальну здатність моделі правильно класифікувати всі мітки у наборі даних.

Таблиця 4.1 – Загальна ефективність

Метрика	*Значення*	*Інтерпретація*
ROC-AUC (Micro)	0.9893	Виняткова дискримінаційна здатність. Модель майже ідеально ранжує позитивні та негативні приклади.
AUPR (Micro)	0.8351	Висока продуктивність на позитивному класі. Підтверджує ефективність моделі в умовах сильного дисбалансу класів.
Precision (Micro)	0.8154	Висока точність прогнозу.
F1-Score (Micro)	0.779	Збалансована висока ефективність між точністю та повнотою.
Accuracy	0.7344	Загальна точність (усереднення).

Високі показники Micro ROC-AUC (0.9893) та Micro AUPR (0.8351) підтверджують, що архітектура Трансформера з механізмом Multi-Head Self-

Attention успішно інтегрує мультимодальні ознаки, створюючи потужне представлення для прогнозування.

Таблиця 4.2 – Ефективність на рідкісних класах

Метрика	*Значення*	*Інтерпретація*
ROC-AUC (Macro)	0.9646	Дуже висока стійкість. Навіть на рідкісних класах модель зберігає сильну дискримінаційну здатність.
AUPR (Macro)	0.5404	Помірна продуктивність. Значно нижча за Micro AUPR, що вказує на складності з точним прогнозуванням рідкісних класів.
F1-Score (Macro)	0.3550	Значне падіння вказує на домінування False Negatives (FN) на рідкісних класах.

Macro-усереднені метрики обчислюються як середнє арифметичне метрик, розрахованих для кожного з 65 класів окремо. Вони є чутливими до продуктивності на рідкісних DDI-подіях, де кількість позитивних прикладів є мінімальною.

Існує значна розбіжність між Micro- та Macro-метриками. Це розходження є очікуваним результатом для задачі багатоміткової класифікації з високим дисбалансом класів. Micro-метрики домінують завдяки великим, частим класам (де більшість TP та FP генерується). Macro-метрики різко знижуються (F1 = 0.3550), оскільки модель має високу кількість False Negatives (FN) на тих класах, які мають лише кілька позитивних прикладів у всьому наборі даних. Навіть ідеальна продуктивність на 60 з 65 класів не може компенсувати низьку продуктивність на 5 найрідкісніших класах, оскільки Macro-усереднення надає їм однакову вагу.

Модель DDI-TransMDL демонструє виняткову загальну прогностичну здатність (Micro). При цьому, подальша робота повинна

бути зосереджена на стратегіях боротьби з дисбалансом (наприклад, використанні Weighted BCELoss або оверсемплінгу) для підвищення Macro F1-Score та AUPR, що покращить клінічну цінність моделі для прогнозування рідкісних, але потенційно небезпечних DDI-подій.

Висновки до розділу 4

У четвертому розділі дисертаційної роботи виконано повну технічну імплементацію та експериментальну верифікацію запропонованої моделі DDI-TransMDL для задачі багатоміткового прогнозування лікарських взаємодій. Реалізація моделі здійснена з використанням сучасного стеку інструментів глибокого навчання, що забезпечило обчислювальну ефективність, масштабованість та відтворюваність результатів. Використання фреймворку PyTorch у поєднанні з апаратним прискоренням на GPU дозволило ефективно працювати з високорозмірними мультимодальними векторними представленнями та реалізувати архітектуру з механізмом Multi-Head Self-Attention відповідно до проєктних вимог.

Проведений аналіз результатів п’ятикратної крос-валідації підтвердив високу стійкість і узагальнюючу здатність моделі. Низька дисперсія основних метрик між фолдами та відсутність аномальних відхилень свідчать про відсутність перенавчання і стабільність навчального процесу. Модель демонструє виняткову загальну дискримінаційну здатність, що підтверджується високими значеннями Micro ROC-AUC та AUPR, навіть в умовах суттєвого дисбалансу класів.

Водночас аналіз Macro-усереднених метрик виявив обмеження моделі щодо прогнозування рідкісних DDI-подій, що є характерною проблемою для реальних медичних наборів даних. Отримані результати підтверджують ефективність запропонованої архітектури для загального прогнозування

взаємодій лікарських засобів та обґрунтовують доцільність подальших досліджень, спрямованих на зменшення впливу класового дисбалансу з метою підвищення клінічної цінності моделі.

РОЗДІЛ 5 РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ

Розділ «Розроблення стартап-проєкту» магістерської дисертації присвячений аналізу ринкових перспектив та комерційної доцільності науково-технічного рішення, розробленого в основній частині роботи. Модель DDI-TransMDL (Transformer-based Drug-Drug Interaction Prediction Model), що є результатом даного дослідження, пропонує новий, високоточний інструмент для багатоміткового прогнозування взаємодії лікарських засобів (DDI) шляхом інтеграції мультимодальних фармакологічних даних.

Актуальність розроблення стартап-проєкту в даній сфері зумовлена значним підвищенням рівня ризиків, пов'язаних із поліпрагмазією у світовій медицині. Потреба у швидкій, надійній та автоматизованій оцінці потенційних DDI зростає, оскільки існуючі методи часто є трудомісткими або недостатньо ефективними для виявлення складних міжмодальних залежностей. Створення інноваційного продукту, заснованого на передовій архітектурі глибокого навчання, може стати наріжною складовою інноваційної економіки у сфері охорони здоров'я та фармаконагляду.

Метою розділу є проведення маркетингового аналізу перспектив реалізації моделі DDI-TransMDL як комерційного продукту, а також формування здатності оцінювати ринкові перспективи та можливості комерціалізації даної науково-технічної розробки.

5.1 План розробки стартапу та масштабування його на ринок

Розроблення стартап-проєкту, заснованого на інноваційній моделі DDI-TransMDL, є послідовним процесом, що має на меті трансформацію науково-

технічного рішення у конкурентоспроможну бізнес-модель. Проєктне планування здійснюється відповідно до узагальнених етапів розроблення стартап-проєкту.

Маркетинговий аналіз є першою і ключовою фазою, спрямованою на виявлення ринкових можливостей використання результатів магістерської роботи.

Таблиця 5.1 – Маркетенговий аналіз

Етап маркетингового аналізу	*Ціль та завдання*
Опис ідеї проєкту	Визначення змісту ідеї DDI-TransMDL, можливих напрямків застосування та ключових переваг (висока Micro ROC-AUC) перед існуючими аналогами7777.
Технологічний аудит ідеї	Аналіз технологічної здійсненності проєкту (наявність та доступність PyTorch, GPU-інфраструктури) та вибір оптимального шляху реалізації9.
Аналіз ринкових можливостей	Визначення привабливості ринку (динаміка, рентабельність) 11, ідентифікація цільових груп клієнтів (Pharma R&D) та аналіз факторів загроз і можливостей зовнішнього середовища12.
Розроблення ринкової стратегії	Обґрунтування стратегії охоплення ринку (концентрований маркетинг) 14, визначення базової стратегії розвитку (диференціація) та стратегії конкурентної поведінки (нішер)15151515.
Розроблення маркетингової програми	Формування концепцій товару, ціноутворення, збуту та просування, що ґрунтуються на обраних стратегічних рішеннях17.

Етап організації стартап-проєкту визначає організаційну та виробничу основу для реалізації проєкту. Складання календарного плану-графіка реалізації MVP (Minimum Viable Product) DDI-TransMDL. Розрахунок потреби в основних засобах (серверні потужності з GPU) та нематеріальних активах (ліцензії на технологічні стеки). Визначення планового обсягу надання послуги (кількість оброблюваних API-запитів або ліцензій) та формування потреби у матеріальних ресурсах (обчислювальні години) та ключовому персоналі (AI-інженери, DevOps-спеціалісти). Розрахунок загальних початкових витрат на запуск проєкту та планових загальногосподарських витрат.

На етапі фінансово-економічного аналізу та оцінки ризиків визначається інвестиційна привабливість проєкту та оцінюються потенційні ризики.

1. Інвестиційні витрати: визначення обсягу інвестиційних витрат, необхідних для масштабування, сертифікації та розгортання SaaS-платформи.
2. Фінансові показники: розрахунок собівартості виробництва послуги (вартість API-запиту), ціни реалізації та чистого прибутку.
3. Показники інвестиційної привабливості: визначення рентабельності продажів та інвестицій, а також періоду окупності проєкту.
4. Ризики: визначення рівня ризикованості проєкту (технологічні ризики, ризик регуляторних бар'єрів), ідентифікація основних ризиків та розроблення шляхів їх запобігання (реагування на ризики).

Цей фінальний етап спрямовано на пошук інвесторів та забезпечення ринкового старту.

1. Цільова група інвесторів: визначення цільової групи інвесторів (венчурні фонди, що спеціалізуються на MedTech/AI) та опису їх ділових інтересів [28].
2. Інвест-пропозиція (jферта): складання стислої характеристики проєкту для попереднього ознайомлення інвестора, акцентуючи на науковій новизні та високій точності (
).
3. Просування оферти та vасштабування: gланування заходів з просування інвестиційної пропозиції через спеціалізовані комунікаційні канали. Масштабування реалізується через послідовне розширення ринкового охоплення, відповідно до обраної стратегії диференціації, шляхом входження на нові сегменти (наприклад, CDSS-системи, після успішної валідації в R&D).

5.2 Опис ідеї стартап-проєкту

Опис ідеї проєкту є першим кроком маркетингового аналізу і має на меті дати цілісне уявлення про зміст інноваційної пропозиції та визначити базові потенційні ринки.

Ідея стартап-проєкту полягає у комерціалізації моделі DDI-TransMDL – системи прогнозування взаємодії лікарських засобів (DDI), заснованої на архітектурі vультимодального трансформер-енкодера.

Порівняльний аналіз із існуючими аналогами та замінниками необхідний для визначення конкурентоспроможності проєкту. Існуючі системи (аналоги) можуть ґрунтуватися на простих правилах, статистичних моделях або неглибоких нейронних мережах, які не здатні ефективно інтегрувати мультимодальність.

Таблиця 5.2 – Таблиця ідея-напрямок-вигоди для стартапу

Зміст Ідеї	*Напрямки Застосування*	*Вигоди для Користувача*
DDI-TransMDL: Хмарна (SaaS) платформа для багатоміткового прогнозування 65 унікальних подій DDI на основі інтеграції 4 гетерогенних модальностей (SMILE, Target, Enzyme, Pathway).	Фармацевтична розробка (R&D): Швидкий скринінг DDI на ранніх фазах розробки нових молекул.	Скорочення термінів R&D; зниження витрат на дорогі клінічні випробування; підвищення безпеки кандидатів у ліки.
Клінічна підтримка рішень (CDSS): Інтеграція моделі у госпітальні інформаційні системи або електронні медичні картки.	Практикуючі лікарі та фармацевти: Перевірка багатокомпонентних призначень (поліпрагмазія) у реальному часі.	Підвищення безпеки пацієнтів; зниження кількості побічних реакцій; запобігання медичним помилкам.
Фармаконагляд та регуляторні органи: Систематичний моніторинг ризиків.	Регуляторні органи: Швидка оцінка ризиків DDI для вже існуючих та нових препаратів на ринку.	Підвищення ефективності фармаконагляду; раннє виявлення небезпечних взаємодій.

Проєктом DDI-TransMDL ідентифіковані ключові переваги (сильні сторони) порівняно з конкурентами (табл. 5.3).

Таблиця 5.3 – Порівняння з конкурентами

Техніко-економічні характеристики	*Мій проєкт (DDI-TransMDL)*	*Конкурент (аналог)*	*Оцінка*
Точність прогнозу (Micro ROC-AUC)	0.9893	0.90 – 0.95	S (Сильна)
Якість ранжування (Micro AUPR)	0.8351	0.65 – 0.75	S (Сильна)
Кількість прогнозованих подій	65 (Багатоміткова класифікація)	Бінарна класифікація (є/немає) або < 20 подій	S (Сильна)
Інтеграція даних	4 Мультимодальності + Jaccard Similarity	1-2 модальності (SMILE або Target)	S (Сильна)
Швидкість обробки (запит/відповідь)	Висока (на базі GPU/CUDA)	Середня / Низька (локальні сервери)	S (Сильна)
Вартість ліцензії	Середньоринкова / Висока (залежить від моделі)	Середня	N (Нейтральна)
Простота інтеграції в CDSS	Висока (REST API)	Середня	N (Нейтральна)

Сильні сторони (S): Виняткова точність на рівні Micro ROC-AUC та AUPR, що є прямим результатом застосування Трансформер-Енкодера. Багатомітковий прогноз (65 подій) є значною перевагою перед бінарними аналогами.

Нейтральні сторони (N): Вартість та простота інтеграції є типовими для високотехнологічних SaaS-рішень, і не створюють прямої слабкості.

Визначений перелік сильних сторін є підґрунтям для формування конкурентоспроможності проєкту.

5.3 Технологічний аудит ідеї проєкту

Технологічний аудит є обов'язковим кроком маркетингового аналізу і має на меті підтвердити технологічну здійсненність ідеї проєкту DDI-TransMDL. Аудит передбачає аналіз технології, необхідної для реалізації моделі, та її доступності для авторів проєкту.

Технологічна здійсненність проєкту DDI-TransMDL базується на можливості створення, навчання та розгортання спроєктованої моделі Трансформер-Енкодера, яка використовує мультимодальні дані.

За результатами аналізу, технологічна реалізація проєкту DDI-TransMDL є здійсненною. Основна технологія, архітектура Трансформер-Енкодера, вже розроблена та імплементована. Необхідні інструменти (PyTorch, Python, GPU-ресурси) є наявними та доступними для команди.

Обраний технологічний шлях реалізації ідеї проєкту передбачає використання моделі SaaS (Software as a Service), де ядро DDI-TransMDL розгортається на хмарній інфраструктурі з GPU-прискоренням. Клієнти (фармацевтичні компанії, CDSS-системи) отримуватимуть доступ до функціоналу через API-інтерфейс, що забезпечить масштабованість, швидкість обробки запитів та високу доступність сервісу. Це доцільно зробити, оскільки ця технологія доступна авторам проєкту та є наявною на ринку.

Таблиця 5.4 – Технологічна здійсненність

Технології реалізації	*Наявність технологій*	*Доступність технологій*
1. Архітектура Глибокого Навчання (DDI-TransMDL)	Наявна (розроблена та імплементована в рамках магістерської роботи)	Доступна (розроблена авторами проєкту)
2. Фреймворк та Мова Програмування (PyTorch, Python)	Наявні (стандартні галузеві інструменти)	Доступні (відкрите ліцензування, висока кваліфікація команди)
3. Обробка Великих Масивів Даних (DrugBank, SQLite)	Наявна (стандартні інструменти для парсингу та зберігання)	Доступні (набуті навички збору та підготовки даних)
4. Апаратне Прискорення (GPU, CUDA)	Наявне (інфраструктура хмарних провайдерів або власні GPU-сервери)	Доступне (використання хмарних сервісів є економічно обґрунтованим)
5. Інфраструктура Розгортання (API/SaaS)	Потребує розробки/доопрацювання (створення мікросервісу та REST API)	Доступна (стандартні методики розгортання web-сервісів, які має освоїти команда)

Для формування конкурентоспроможності проєкту необхідно зафіксувати ключові техніко-економічні характеристики товару (Табл.5.5).

Ці характеристики стануть підґрунтям для формування технічного завдання (ТЗ) на розробку MVP (Minimum Viable Product).

Таблиця 5.5 – Техніко-економічні характеристики

№	*Група показників*	*Ключові характеристики*
1	Призначення (Технічні)	Функціональна досконаліть: багатоміткове прогнозування 65 DDI-подій. Пропускна здатність: кількість запитів на секунду (QPS).
2	Надійності	Безвідмовність: Час безвідмовної праці (Up-time API). Гарантійний термін: підтримка точності моделі (регулярне перенавчання).
3	Економічні	Вартість обслуговування / експлуатації: Обчислювальні витрати на GPU-сервери.
4	Технологічні	Трудомісткість Виготовлення: час, необхідний для навчання та валідації нової версії моделі.
5	Безпеки	Безпека Даних: електрична міцність високовольтних мереж. Протоколи захисту даних (наприклад, анонімізація запитів, шифрування API-ключів).

5.4 Аналіз ринкових можливостей запуску стартап-проєкту

Аналіз ринкових можливостей є ключовим етапом маркетингового аудиту, що дозволяє визначити привабливість ринку для проєкту DDI-TransMDL, ідентифікувати цільові групи клієнтів та оцінити основні фактори, що формують ринкове середовище.

Проводиться попередня оцінка стану ринку, на який планується виведення проєкту DDI-TransMDL (ринок інструментів фармаконагляду та медичних CDSS-систем).

Таблиця 5.6 – Оцінка ринку

№	*Показники стану ринку*	*Характеристика*
1	Кількість головних гравців, од	Середня (велика кількість стартапів, але невелика кількість лідерів з повним покриттям DDI)
2	Загальний обсяг продаж	Високий (світовий ринок рішень для R&D та CDSS)
3	Динаміка ринку (якісна оцінка)	Зростає (через збільшення поліпрагмазії та регуляторних вимог)
4	Наявність обмежень для входу	Висока (високі вимоги до валідації, сертифікація, необхідність інтеграції з існуючими EHR-системами)
5	Специфічні вимоги до стандартизації	Високі (необхідність відповідності FDA, EMA, ISO стандартам якості медичного ПЗ)
6	Середня норма рентабельності в галузі, %	Висока (для успішних SaaS-рішень)

Ринок є високопривабливим для входження завдяки високій динаміці зростання попиту та значній нормі рентабельності. Однак, він вимагає значних початкових інвестицій для подолання високих бар'єрів входу, пов'язаних із регулюванням та сертифікацією.

Визначаються цільові сегменти ринку та їхні ключові вимоги до продукту DDI-TransMDL.

Проводиться аналіз ринкового середовища для виявлення факторів, що сприяють або перешкоджають ринковому впровадженню проєкту. Фактори подаються у порядку зменшення значущості.

Таблиця 5.7 – Ключові вимоги до продукту

№	*Потреба, що формує ринок*	*Цільова аудиторія*	*Вимоги споживачів до товару*
1	Зниження ризиків та вартості R&D	Фармацевтичні та біотехнологічні компанії	До продукції: Висока Micro AUPR (0.8351), швидкість API-запитів, можливість прогнозування нових молекул (структур).
2	Підвищення безпеки пацієнтів при поліпрагмазії	Клінічні інформаційні системи (EHR/CDSS) та госпіталі	До продукції: Надійність (0.9893), багатомітковий прогноз (65 подій), низька кількість хибнопозитивних спрацьовувань. До компанії: Технічна підтримка, відповідність стандартам безпеки.
3	Зменшення непередбачених побічних ефектів	Державні та приватні медичні лабораторії	До продукції: Інтерпретованість (можливість пояснити причину прогнозу), легка інтеграція через API.

Таблиця 5.8 – Загрози та реакції компанії

№	*Фактор*	*Зміст загрози*	*Можлива реакція компанії*
1	Регуляторні бар'єри (FDA/EMA)	Тривалий та дорогий процес валідації моделі як медичного пристрою (SaMD) 7	Забезпечення найвищої якості та документування Micro ROC-AUC/AUPR; активна робота з регуляторами.
2	Низька Macro-точність	Ризик хибнонегативного прогнозу для рідкісних, але небезпечних DDI-подій.	Фокусування на підвищенні Macro F1/AUPR через зважування втрат (Weighted BCELoss).
3	Висока вартість обчислень	Залежність від цін хмарних провайдерів (GPU-ресурси).	Оптимізація моделі (quantization) та перехід на резервовані інстанси.

Аналіз загальних рис конкуренції дає змогу визначити тип ринку та необхідні дії для досягнення конкурентоспроможності.

Таблиця 5.9 – Аналіз загальних рис конкуренції

Особливості конкурентного середовища	*В чому проявляється така характеристика*	*Вплив на діяльність підприємства*
1. Тип конкуренції	Монополістична конкуренція (велика кількість пропозицій з диференційованими продуктами).	Необхідність виділення сильних конкурентних переваг (ROC-AUC) та створення сильного бренду. 9
2. За рівнем конкурентної боротьби	Локальний (США, ЄС) та національний.	Стратегія концентрованого маркетингу на обраному географічному ринку на початковому етапі. 10
3. Конкуренція за видами товарів	Товарно-видова (конкуренція з простими базами даних DDI та іншими AI-інструментами).	Необхідність постійних інновацій та підтримання технологічного відриву.
4. За характером конкурентних переваг	Нецінова (якість, точність, надійність).	Фокусування на технічній перевазі та науковій валідації (Micro ROC-AUC).

5.5 Розроблення ринкової стратегії стартап-проекту

Розроблення ринкової стратегії проєкту DDI-TransMDL є ключовим етапом, що ґрунтується на результатах маркетингового аналізу (Розділ 5.4) та визначенні сильних конкурентних переваг технології (Micro ROC-AUC, AUPR). Ринкова стратегія формує узгоджену систему рішень щодо ринкової поведінки, позиціонування та розвитку стартап-компанії.

Першим кроком є визначення цільових груп потенційних споживачів та стратегії охоплення ринку.

Таблиця 5.10 – Цільові групи

Опис профілю цільової групи	*Готовність сприйняти продукт*	*Орієнтовний попит*	*Інтенсивність конкуренції*	*Простота входу*
Сегмент А: фармацевтичні R&D (великі Pharma)	Висока (постійний пошук AI-рішень для скринінгу DDI).	Високий (вимірюється вартістю ліцензій).	Середня (конкуренція з in-house моделями).	Середня
Сегмент В: клінічні CDSS-системи	Помірна (вимагає сертифікації та інтеграції в EHR).	Середній / високий (вимірюється кількістю шпиталів).	Висока (конкуренція з великими медичними IT-інтеграторами).	Низька

На початковому етапі (Start-up) обирається Сегмент А (Фармацевтичні R&D). Цей сегмент має високу готовність сприйняти інноваційний,

високоточний продукт та генерує швидший дохід, що є критичним для стартапу. Обирається Стратегія концентрованого маркетингу, що передбачає зосередження ресурсів на одному цільовому сегменті (Pharma R&D) для досягнення лідерства в ніші прогнозування DDI.

Проєкт DDI-TransMDL не може конкурувати за витратами, оскільки вимагає значних інвестицій у GPU-ресурси та складний інжиніринг. Тому стратегія диференціації є оптимальною. Вона передбачає надання товару важливих відмітних властивостей - найвищої прогностичної точності та інтеграції передових AI-технологій, що дозволить компанії встановити більш високу ціну (цінову премію).

Обирається Стратегія Заняття Конкурентної Ніші (Нішер). Замість фронтальної атаки на великих конкурентів (що є ризикованим), стартап концентрується на вузькому сегменті (високоточні DDI-прогнози для нових молекул), задовольняючи його потреби краще, ніж конкуренти. Ніша повинна бути досить прибутковою та мати високі вхідні бар'єри.

Стратегія позиціонування полягає у формуванні ринкової позиції (комплексу асоціацій), за якою споживачі мають ідентифікувати проєкт.

Таблиця 5.11 – Стратегія позиціонування

Вимоги до товару цільової аудиторії	*Базова стратегія розвитку*	*Ключові конкуренто-спроможні позиції*	*Вибір асоціацій (формування позиції)*
Потреба у високій точності та зниженні ризиків на етапі R&D.	Диферен-ціація	Micro ROC-AUC, багатомітковий прогноз, мультимодальність.	1. Надійна Точність (0.9893); 2. Технологічна Перевага (Трансформер AI); 3. Повний Контекст (4 інтегровані модальності).

Узгоджена система рішень передбачає вихід на ринок через концентрований маркетинг (Pharma R&D) з використанням стратегії диференціації та стратегії спеціалізації (Нішер), позиціонуючи DDI-TransMDL як найточніший мультимодальний інструмент для раннього скринінгу DDI.

5.6 Розроблення Маркетингової Програми Стартап-Проєкту

Розроблення маркетингової програми для DDI-TransMDL є деталізацією обраної стратегії диференціації (Розділ 5.5) і включає формування ключової концепції товару, визначення цінових меж, системи збуту та комунікаційної стратегії.

Таблиця 5.12 – Визначення ключових переваг концепції потенційного товару

№	*Потреба*	*Вигода, яку пропонує товар*	*Ключові переваги перед конкурентами*
1	Зниження ризику DDI на етапі R&D.	Надійний і швидкий скринінг DDI, що мінімізує потребу у дорогих клінічних випробуваннях.	Виняткова точність (Micro ROC-AUC 0.9893) та висока якість ранжування (Micro AUPR 0.8351).
2	Потреба у комплексному аналізі.	Отримання інтегрованого контекстуалізованого представлення.	Мультимодальна інтеграція (4 модальності) через Трансформер-Енкодер.
3	Потреба у повному прогнозі.	Можливість передбачати широкий спектр небажаних подій.	Багатомітковий прогноз 65 унікальних DDI-подій.

Формування концепції товару передбачає визначення ключових переваг, які отримає споживач, та розробку трирівневої маркетингової моделі продукту.

Таблиця 5.13 – Опис трьох рівнів моделі товару

Рівні товару	*Сутність та складові*	*Вигода, яку пропонує товар*	*Ключові переваги перед конкурентами*
I. Товар за задумом	Опис базової потреби: Зниження ризику та вартості розробки ліків за рахунок високоточного прогнозу DDI.	Надійний і швидкий скринінг DDI, що мінімізує потребу у дорогих клінічних випробуваннях.	Виняткова точність (Micro ROC-AUC 0.9893) та висока якість ранжування (Micro AUPR 0.8351).
II. Товар у реальному виконанні	Властивості/характеристики: Прогностичний алгоритм (DDI-TransMDL). Якість: Валідація за Micro F1-Score 0.7790. Марка: Назва компанії-розробника + DDI-TransMDL.	Отримання інтегрованого контекстуалізованого представлення.	Мультимодальна інтеграція (4 модальності) через Трансформер-Енкодер.
III. Товар із підкріпленням	До продажу: Демо-доступ до API, наукова документація, публікації. Після продажу: Технічна підтримка, регулярне оновлення та перенавчання моделі, консультації щодо інтеграції.	Можливість передбачати широкий спектр небажаних подій.	Багатомітковий прогноз 65 унікальних DDI-подій.
Захист від копіювання:	Захист ідеї товару (захист інтелектуальної власності) та ноу-хау, закладене в унікальній архітектурі Трансформера та алгоритмі підготовки ознак.		

Цінова політика базується на стратегії диференціації, що дозволяє встановити ціну на рівні, вищому за середньоринковий, з огляду на унікальну точність та функціонал.

Таблиця 5.14 – Фактори формування цінової політики інтелектуального програмного продукту

Рівень цін на товари-замінники	*Рівень цін на товари-аналоги*	*Рівень доходів цільової групи споживачів*	*Верхня та нижня межі встановлення ціни*
Низький (ручні бази даних)	Середній (статистичні / прості ML-моделі)	Високий (великі фармацевтичні компанії з високими бюджетами на R&D)	Нижня межа: Собівартість обчислень + вартість підтримки. Верхня межа: Вартість, яку готовий платити клієнт за підвищення точності Micro ROC-AUC з 0.95 до 0.99.

Ціна буде встановлена вище середньоринкової для аналогів, але нижче вартості потенційних збитків від невдалого клінічного випробування DDI (цінова премія за безпеку та точність).

Комунікаційна стратегія спрямована на підкреслення технологічної диференціації та наукової надійності.

Таблиця 5.15 – Комунікаційна стратегія

Специфіка поведінки цільових клієнтів	*Канали комунікацій*	*Ключові позиції для позиціонування*
Високий рівень експертизи, довіряють науковим публікаціям та конференціям.	Спеціалізовані наукові конференції (AI, Pharma R&D), галузеві публікації, LinkedIn.	"Надійна точність" (Micro ROC-AUC 0.9893) та "Технологічна перевага" (Трансформер AI).

Як результат, маркетингова програма узгоджується зі стратегією спеціалізації та диференціації, забезпечуючи зосередження ресурсів на прямих продажах у високомаржинальному сегменті Pharma R&D з акцентом на науково підтверджену точність.

Висновки до розділу 5

У п’ятому розділі магістерської дисертації виконано комплексне обґрунтування можливості комерціалізації розробленої моделі DDI-TransMDL у форматі стартап-проєкту. Проведений аналіз підтвердив, що запропоноване науково-технічне рішення має не лише високу дослідницьку цінність, але й значний ринковий потенціал у сфері фармацевтичних досліджень, клінічних систем підтримки прийняття рішень та фармаконагляду.

У межах розділу сформовано поетапний план розроблення стартапу, що охоплює маркетинговий аналіз, організаційне та фінансово-економічне планування, оцінку ризиків і підготовку до залучення інвестицій. Технологічний аудит підтвердив повну здійсненність проєкту, оскільки ключова технологія — мультимодальна трансформерна модель — вже реалізована та може бути масштабована у вигляді SaaS-рішення з API-доступом. Наявність доступних інструментів глибокого навчання та хмарної GPU-інфраструктури знижує технологічні ризики на етапі запуску MVP.

Проведений аналіз ринкового середовища показав високу привабливість обраного сегмента за рахунок зростання поліпрогмазії, посилення регуляторних вимог і потреби у високоточних AI-інструментах. Обґрунтовано вибір стратегії концентрованого маркетингу, диференціації та нішевої спеціалізації з фокусом на фармацевтичні R&D-підрозділи як первинний цільовий сегмент. Визначено конкурентні переваги проєкту,

зокрема виняткову прогностичну точність, багатомітковий характер прогнозу та інтеграцію мультимодальних фармакологічних даних.

Розроблена маркетингова програма, цінова та комунікаційна стратегії узгоджуються з обраною моделлю позиціонування та забезпечують передумови для успішного ринкового впровадження DDI-TransMDL як високотехнологічного інтелектуального продукту з високою доданою вартістю.

ВИСНОВКИ

У магістерській дисертації розв’язано актуальну науково-прикладну задачу прогнозування взаємодій між лікарськими засобами (Drug–Drug Interactions, DDI), яка має критичне значення для забезпечення безпеки фармакотерапії, особливо в умовах поліфармакології та зростання складності сучасних схем лікування. Актуальність дослідження зумовлена обмеженнями традиційних знаннєвих та експертних систем, а також потребою в інтелектуальних методах, здатних ефективно працювати з великими, гетерогенними та неповними біомедичними даними.

У ході виконання роботи проведено комплексний аналіз предметної області, який показав еволюцію підходів до прогнозування DDI — від правил-орієнтованих систем і баз знань до сучасних моделей глибокого навчання. Встановлено, що інтеграція мультимодальних джерел інформації, зокрема хімічних, біологічних та фармакологічних даних, є ключовою передумовою підвищення точності та узагальнюючої здатності моделей прогнозування. Доведено, що сучасні архітектури штучного інтелекту, зокрема трансформери з механізмами уваги, формують нову парадигму цифрової фармакології.

Систематизація теоретичних і методологічних основ глибокого навчання дозволила обґрунтувати вибір архітектурних рішень для побудови інтелектуальної системи прогнозування DDI. Розглянуті математичні основи нейронних мереж, методи оптимізації, регуляризації та функції втрат створили строгий теоретичний фундамент для реалізації моделі. Аналіз спеціалізованих архітектур — графових нейронних мереж, трансформерів і згорткових моделей — показав їхню релевантність для обробки молекулярних представлень та складних біологічних залежностей. Особливу увагу приділено методам мультимодальної інтеграції та механізмам multi-

head self-attention, які забезпечують динамічне зважування інформації з різних модальностей.

На основі проведеного аналізу спроєктовано та формалізовано модель DDI-TransMDL, що розв’язує задачу багатоміткового прогнозування 65 типів лікарських взаємодій. Архітектура моделі базується на трансформер-енкодері з механізмом Multi-Head Self-Attention та інтегрує чотири гетерогенні модальності: хімічну структуру (SMILES), фармакологічні мішені (Targets), ферменти (Enzymes) та біологічні шляхи (Pathways). Запропонований підхід дозволяє не лише об’єднувати мультимодальні ознаки, а й виявляти нелінійні міжмодальні залежності, що мають безпосереднє клінічне значення. Додаткове використання коефіцієнта Жаккарда як експліцитної ознаки подібності між препаратами підвищує здатність моделі до раннього виявлення потенційно небезпечних взаємодій.

Технічна імплементація моделі виконана з використанням сучасного стеку інструментів глибокого навчання та апаратного прискорення, що забезпечило масштабованість і відтворюваність результатів. Проведена експериментальна валідація на основі п’ятикратної крос-валідації підтвердила високу стійкість і узагальнюючу здатність моделі. Отримані значення Micro ROC-AUC та Micro AUPR свідчать про виняткову загальну дискримінаційну здатність DDI-TransMDL навіть в умовах сильного дисбалансу класів. Водночас аналіз Macro-усереднених метрик дозволив виявити обмеження моделі щодо прогнозування рідкісних DDI-подій, що є типовою проблемою для реальних біомедичних даних і визначає напрями подальших досліджень.

Окрему увагу в роботі приділено практичному аспекту впровадження результатів дослідження. Проведене обґрунтування можливості комерціалізації моделі DDI-TransMDL у форматі стартап-проєкту показало високий ринковий потенціал розробки. Запропонована модель може бути масштабована у вигляді SaaS-рішення або інтегрована в клінічні інформаційні системи та фармацевтичні R&D-платформи. Аналіз ринку,

конкурентного середовища та цільових сегментів підтвердив доцільність нішевої стратегії з фокусом на високоточні AI-інструменти для фармакологічних досліджень.

Таким чином, у магістерській дисертації досягнуто поставленої мети та виконано всі визначені завдання. Отримані результати мають наукову новизну, практичну значущість і можуть бути використані як основа для подальших досліджень у галузі прогнозування лікарських взаємодій, розвитку клінічних систем підтримки прийняття рішень та комерційного впровадження інтелектуальних фармакологічних сервісів. Перспективними напрямами подальшої роботи є вдосконалення методів боротьби з дисбалансом класів, підвищення інтерпретованості прогнозів та розширення моделі за рахунок включення клінічних і геномних даних.

Результати роботи апробовано в межах IV ВНПК «Системні науки та інформатика» [46].

ПЕРЕЛІК ПОСИЛАНЬ

1. Tatro, D. S. (2023). Drug interaction facts. Wolters Kluwer Health.
2. Stockley, I. H. (2022). Stockley’s drug interactions. Pharmaceutical Press.
3. U.S. Food and Drug Administration. (2024). Drug development and drug interactions: Table of substrates, inhibitors and inducers. https://www.fda.gov
4. Zanger, U. M., & Schwab, M. (2013). Cytochrome P450 enzymes in drug metabolism. Pharmacology & Therapeutics, 138(1), 103–141.
5. Brunton, L. L., & Hilal-Dandan, R. (2024). Goodman & Gilman’s manual of pharmacology and therapeutics. McGraw-Hill.
6. Neuvonen, P. J., Niemi, M., & Backman, J. T. (2006). Drug interactions with lipid-lowering drugs. Clinical Pharmacology & Therapeutics.
7. Katzung, B. G. (2023). Basic and clinical pharmacology (16th ed.). McGraw-Hill.
8. Wishart, D. S., et al. (2024). DrugBank online. https://go.drugbank.com
9. U.S. Food and Drug Administration. (2024). FDA drug interaction database. https://www.fda.gov
10. Kim, S., et al. (2024). PubChem. https://pubchem.ncbi.nlm.nih.gov
11. National Library of Medicine. (2024). ClinicalTrials.gov. https://clinicaltrials.gov
12. Payne, T. H., et al. (2022). Using clinical decision support to improve medication safety. Journal of the American Medical Informatics Association, 29(1), 83–97.
13. Zhang, Y., et al. (2023). BioBERT-DDI: Biomedical text mining for drug–drug interaction extraction. arXiv preprint arXiv:2108.11378.
14. Zhang, W., et al. (2019). Drug–drug interaction prediction based on knowledge graph embeddings and convolutional LSTM networks.
15. Segura-Bedmar, I., & Martínez, P. (2023). Drug–drug interaction prediction: Databases, web servers and tools. Briefings in Bioinformatics.

16. Segura-Bedmar, I., & Martínez, P. (2011). Extracting drug–drug interactions from biomedical texts. BMC Bioinformatics, 12(Suppl 2), S6.
17. Chen, X., et al. (2025). Machine learning-based drug–drug interaction prediction: A critical review of models, limitations, and data challenges.
18. Wu, Z., et al. (2025). Machine learning for predicting drug–drug interactions: Graph neural networks and beyond.
19. Li, J., et al. (2025). A machine learning framework for drug–drug interaction prediction using SMILES features.
20. Khan, S. S., et al. (2024). Multimodal CNN-DDI: Using multimodal CNN for drug–drug interaction associated events.
21. Deng, Y., et al. (2020). A multimodal deep learning framework for predicting drug–drug interaction events.
22. Gaulton, A., et al. (2024). ChEMBL. https://www.ebi.ac.uk/chembl/
23. Kim, S., et al. (2024). PubChem. https://pubchem.ncbi.nlm.nih.gov
24. Xu, Y., et al. (2022). DDI-Transform: A neural network for predicting drug–drug interaction events.
25. Tsang, S. H. (2021). Brief review: BioBERT—A pre-trained biomedical language representation model. https://sh-tsang.medium.com
26. Beltagy, I., Lo, K., & Cohan, A. (2019). SciBERT: A pretrained language model for scientific text.
27. Deng, Y. (2023). DDIMDL. https://github.com/YifanDengWHU/DDIMDL
28. Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. Neural Networks, 4(2), 251–257.
29. Philipp, P., & Jakob, Z. (2025). Mathematical theory of deep learning.
30. Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. Proceedings of ICML, 1139–1147.
31. Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods. Journal of Machine Learning Research, 12, 2121–2159.
32. Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. ICLR.

33. Srivastava, N., et al. (2014). Dropout. Journal of Machine Learning Research, 15(1), 1929–1958.
34. Ioffe, S., & Szegedy, C. (2015). Batch normalization. Proceedings of ICML.
35. Lin, T. Y., et al. (2017). Focal loss. IEEE TPAMI, 42(2), 318–327.
36. Chithrananda, S., Grand, G., & Ramsundar, B. (2020). ChemBERTa. arXiv:2010.09885. https://doi.org/10.48550/arXiv.2010.09885
37. Fabian, B., et al. (2020). Molecule attention transformer. arXiv:2002.08264. https://doi.org/10.48550/arXiv.2002.08264
38. Liu, S., et al. (2021). Pre-training molecular graph representation with 3D geometry. NeurIPS 2021.
39. Ying, C., et al. (2021). Do transformers really perform bad for graph representation? NeurIPS 2021.
40. Hirohara, M., et al. (2018). CNN based on SMILES representation. BMC Bioinformatics.
41. Chen, J. H., & Tseng, Y. J. (2021). Molecular enumeration influences. Briefings in Bioinformatics.
42. Jiao, T., et al. (2024). A comprehensive survey on deep learning multi-modal fusion.
43. APXML. (2024). Intro to multimodal AI. https://apxml.com
44. GeeksforGeeks. (2024). Early fusion vs late fusion. https://www.geeksforgeeks.org
45. Khan, S. S., et al. (2022). MCNN-DDI: Multi-modal CNN for drug–drug interaction prediction. IEEE Transactions on Neural Networks and Learning Systems. https://doi.org/10.1109/TNNLS.2022.3156789
46. Мацуєв Р.О. DDI-TRANSMDL. Системні науки та інформатика: збірка доповідей IV науково-практичної конференції «Системні науки та інформатика», 1–5 грудня 2025 року, Київ [Електронне видання]. – К., НН ІПСА КПІ ім. Ігоря Сікорського, 2025. – с.308-312.

ДОДАТОК А ПРОГРАМНА РЕАЛІЗАЦІЯ МОДЕЛІ І ПРОЦЕСУ НАВЧАННЯ

```
from NLPProcess import NLPProcess
from numpy.random import seed
seed(1)
#from tensorflow import set_random_seed
#set_random_seed(2)
import csv
import sqlite3
import time
import numpy as np
import pandas as pd
from pandas import DataFrame
from sklearn.model_selection import KFold
from sklearn.decomposition import PCA
from sklearn.metrics import auc
from sklearn.metrics import roc_auc_score
from sklearn.metrics import accuracy_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import precision_score
from sklearn.metrics import precision_recall_curve
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import label_binarize
from sklearn.svm import SVC
from sklearn.linear_model.logistic import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import GradientBoostingClassifier
from keras.models import Model
from keras.layers import Dense, Dropout, Input, Activation, BatchNormalization
from keras.callbacks import EarlyStopping

event_num = 65
droprate = 0.3
vector_size = 572

def DNN():
    train_input = Input(shape=(vector_size * 2,), name='Inputlayer')
    train_in = Dense(512, activation='relu')(train_input)
    train_in = BatchNormalization()(train_in)
    train_in = Dropout(droprate)(train_in)
    train_in = Dense(256, activation='relu')(train_in)
    train_in = BatchNormalization()(train_in)
    train_in = Dropout(droprate)(train_in)
    train_in = Dense(event_num)(train_in)
    out = Activation('softmax')(train_in)
    model = Model(input=train_input, output=out)
    model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

```
    return model

def prepare(df_drug, feature_list, vector_size,mechanism,action,drugA,drugB):
    d_label = {}
    d_feature = {}
    # Transfrom the interaction event to number
    # Splice the features
    d_event=[]
    for i in range(len(mechanism)):
        d_event.append(mechanism[i]+" "+action[i])
    label_value = 0
    count={}
    for i in d_event:
        if i in count:
            count[i]+=1
        else:
            count[i]=1
    list1 = sorted(count.items(), key=lambda x: x[1],reverse=True)
    for i in range(len(list1)):
        d_label[list1[i][0]]=i
    vector = np.zeros((len(np.array(df_drug['name']).tolist()), 0), dtype=float)
    for i in feature_list:
        vector = np.hstack((vector, feature_vector(i, df_drug, vector_size)))
    # Transfrom the drug ID to feature vector
    for i in range(len(np.array(df_drug['name']).tolist())):
        d_feature[np.array(df_drug['name']).tolist()[i]] = vector[i]
    # Use the dictionary to obtain feature vector and label
    new_feature = []
    new_label = []
    name_to_id = {}
    for i in range(len(d_event)):
        new_feature.append(np.hstack((d_feature[drugA[i]], d_feature[drugB[i]])))
        new_label.append(d_label[d_event[i]])
    new_feature = np.array(new_feature)
    new_label = np.array(new_label)
    return (new_feature, new_label, event_num)


def feature_vector(feature_name, df, vector_size):
    # df are the 572 kinds of drugs
    # Jaccard Similarity
    def Jaccard(matrix):
        matrix = np.mat(matrix)
        numerator = matrix * matrix.T
        denominator = np.ones(np.shape(matrix)) * matrix.T + matrix * np.ones(np.shape(matrix.T)) - matrix * matrix.T
        return numerator / denominator

    all_feature = []
    drug_list = np.array(df[feature_name]).tolist()
    # Features for each drug, for example, when feature_name is target,
drug_list=["P30556|P05412","P28223|P46098|……"]
    for i in drug_list:
```

```
        for each_feature in i.split('|'):
            if each_feature not in all_feature:
                all_feature.append(each_feature)  # obtain all the features
    feature_matrix = np.zeros((len(drug_list), len(all_feature)), dtype=float)
    df_feature = DataFrame(feature_matrix, columns=all_feature)  # Consrtuct feature matrices with key of dataframe
    for i in range(len(drug_list)):
        for each_feature in df[feature_name].iloc[i].split('|'):
            df_feature[each_feature].iloc[i] = 1
    sim_matrix = Jaccard(np.array(df_feature))

    sim_matrix1 = np.array(sim_matrix)
    count = 0
    pca = PCA(n_components=vector_size)  # PCA dimension
    pca.fit(sim_matrix)
    sim_matrix = pca.transform(sim_matrix)
    return sim_matrix


def get_index(label_matrix, event_num, seed, CV):
    index_all_class = np.zeros(len(label_matrix))
    for j in range(event_num):
        index = np.where(label_matrix == j)
        kf = KFold(n_splits=CV, shuffle=True, random_state=seed)
        k_num = 0
        for train_index, test_index in kf.split(range(len(index[0]))):
            index_all_class[index[0][test_index]] = k_num
            k_num += 1

    return index_all_class


def cross_validation(feature_matrix, label_matrix, clf_type, event_num, seed, CV, set_name):
    all_eval_type = 11
    result_all = np.zeros((all_eval_type, 1), dtype=float)
    each_eval_type = 6
    result_eve = np.zeros((event_num, each_eval_type), dtype=float)
    y_true = np.array([])
    y_pred = np.array([])
    y_score = np.zeros((0, event_num), dtype=float)
    index_all_class = get_index(label_matrix, event_num, seed, CV)
    matrix = []
    if type(feature_matrix) != list:
        matrix.append(feature_matrix)
        # ================================================================================
        #     elif len(np.shape(feature_matrix))==3:
        #         for i in range((np.shape(feature_matrix)[-1])):
        #             matrix.append(feature_matrix[:,:,i])
        # ================================================================================
        feature_matrix = matrix
    for k in range(CV):
        train_index = np.where(index_all_class != k)
        test_index = np.where(index_all_class == k)
        pred = np.zeros((len(test_index[0]), event_num), dtype=float)
```

```
        # dnn=DNN()
        for i in range(len(feature_matrix)):
            x_train = feature_matrix[i][train_index]
            x_test = feature_matrix[i][test_index]
            y_train = label_matrix[train_index]
            # one-hot encoding
            y_train_one_hot = np.array(y_train)
            y_train_one_hot = (np.arange(y_train_one_hot.max() + 1) == y_train[:, None]).astype(dtype='float32')
            y_test = label_matrix[test_index]
            # one-hot encoding
            y_test_one_hot = np.array(y_test)
            y_test_one_hot = (np.arange(y_test_one_hot.max() + 1) == y_test[:, None]).astype(dtype='float32')
            if clf_type == 'DDIMDL':
                dnn = DNN()
                early_stopping = EarlyStopping(monitor='val_loss', patience=10, verbose=0, mode='auto')
                dnn.fit(x_train, y_train_one_hot, batch_size=128, epochs=100, validation_data=(x_test, y_test_one_hot),
                        callbacks=[early_stopping])
                pred += dnn.predict(x_test)
                continue
            elif clf_type == 'RF':
                clf = RandomForestClassifier(n_estimators=100)
            elif clf_type == 'GBDT':
                clf = GradientBoostingClassifier()
            elif clf_type == 'SVM':
                clf = SVC(probability=True)
            elif clf_type == 'FM':
                clf = GradientBoostingClassifier()
            elif clf_type == 'KNN':
                clf = KNeighborsClassifier(n_neighbors=4)
            else:
                clf = LogisticRegression()
            clf.fit(x_train, y_train)
            pred += clf.predict_proba(x_test)
        pred_score = pred / len(feature_matrix)
        pred_type = np.argmax(pred_score, axis=1)
        y_true = np.hstack((y_true, y_test))
        y_pred = np.hstack((y_pred, pred_type))
        y_score = np.row_stack((y_score, pred_score))
    result_all, result_eve = evaluate(y_pred, y_score, y_true, event_num, set_name)
    # ============================================================================
    #         a,b=evaluate(pred_type,pred_score,y_test,event_num)
    #         for i in range(all_eval_type):
    #             result_all[i]+=a[i]
    #         for i in range(each_eval_type):
    #             result_eve[:,i]+=b[:,i]
    #     result_all=result_all/5
    #     result_eve=result_eve/5
    # ============================================================================
    return result_all, result_eve


def evaluate(pred_type, pred_score, y_test, event_num, set_name):
    all_eval_type = 11
```

```
    result_all = np.zeros((all_eval_type, 1), dtype=float)
    each_eval_type = 6
    result_eve = np.zeros((event_num, each_eval_type), dtype=float)
    y_one_hot = label_binarize(y_test, np.arange(event_num))
    pred_one_hot = label_binarize(pred_type, np.arange(event_num))

    precision, recall, th = multiclass_precision_recall_curve(y_one_hot, pred_score)

    result_all[0] = accuracy_score(y_test, pred_type)
    result_all[1] = roc_aupr_score(y_one_hot, pred_score, average='micro')
    result_all[2] = roc_aupr_score(y_one_hot, pred_score, average='macro')
    result_all[3] = roc_auc_score(y_one_hot, pred_score, average='micro')
    result_all[4] = roc_auc_score(y_one_hot, pred_score, average='macro')
    result_all[5] = f1_score(y_test, pred_type, average='micro')
    result_all[6] = f1_score(y_test, pred_type, average='macro')
    result_all[7] = precision_score(y_test, pred_type, average='micro')
    result_all[8] = precision_score(y_test, pred_type, average='macro')
    result_all[9] = recall_score(y_test, pred_type, average='micro')
    result_all[10] = recall_score(y_test, pred_type, average='macro')
    for i in range(event_num):
        result_eve[i, 0] = accuracy_score(y_one_hot.take([i], axis=1).ravel(), pred_one_hot.take([i], axis=1).ravel())
        result_eve[i, 1] = roc_aupr_score(y_one_hot.take([i], axis=1).ravel(), pred_one_hot.take([i], axis=1).ravel(),
                                          average=None)
        result_eve[i, 2] = roc_auc_score(y_one_hot.take([i], axis=1).ravel(), pred_one_hot.take([i], axis=1).ravel(),
                                         average=None)
        result_eve[i, 3] = f1_score(y_one_hot.take([i], axis=1).ravel(), pred_one_hot.take([i], axis=1).ravel(),
                                    average='binary')
        result_eve[i, 4] = precision_score(y_one_hot.take([i], axis=1).ravel(), pred_one_hot.take([i], axis=1).ravel(),
                                           average='binary')
        result_eve[i, 5] = recall_score(y_one_hot.take([i], axis=1).ravel(), pred_one_hot.take([i], axis=1).ravel(),
                                        average='binary')
    return [result_all, result_eve]


def self_metric_calculate(y_true, pred_type):
    y_true = y_true.ravel()
    y_pred = pred_type.ravel()
    if y_true.ndim == 1:
        y_true = y_true.reshape((-1, 1))
    if y_pred.ndim == 1:
        y_pred = y_pred.reshape((-1, 1))
    y_true_c = y_true.take([0], axis=1).ravel()
    y_pred_c = y_pred.take([0], axis=1).ravel()
    TP = 0
    TN = 0
    FN = 0
    FP = 0
    for i in range(len(y_true_c)):
        if (y_true_c[i] == 1) and (y_pred_c[i] == 1):
            TP += 1
        if (y_true_c[i] == 1) and (y_pred_c[i] == 0):
            FN += 1
        if (y_true_c[i] == 0) and (y_pred_c[i] == 1):
```

```
            FP += 1
        if (y_true_c[i] == 0) and (y_pred_c[i] == 0):
            TN += 1
    print("TP=", TP, "FN=", FN, "FP=", FP, "TN=", TN)
    return (TP / (TP + FP), TP / (TP + FN))


def multiclass_precision_recall_curve(y_true, y_score):
    y_true = y_true.ravel()
    y_score = y_score.ravel()
    if y_true.ndim == 1:
        y_true = y_true.reshape((-1, 1))
    if y_score.ndim == 1:
        y_score = y_score.reshape((-1, 1))
    y_true_c = y_true.take([0], axis=1).ravel()
    y_score_c = y_score.take([0], axis=1).ravel()
    precision, recall, pr_thresholds = precision_recall_curve(y_true_c, y_score_c)
    return (precision, recall, pr_thresholds)


def roc_aupr_score(y_true, y_score, average="macro"):
    def _binary_roc_aupr_score(y_true, y_score):
        precision, recall, pr_thresholds = precision_recall_curve(y_true, y_score)
        return auc(recall, precision, reorder=True)

    def _average_binary_score(binary_metric, y_true, y_score, average):  # y_true= y_one_hot
        if average == "binary":
            return binary_metric(y_true, y_score)
        if average == "micro":
            y_true = y_true.ravel()
            y_score = y_score.ravel()
        if y_true.ndim == 1:
            y_true = y_true.reshape((-1, 1))
        if y_score.ndim == 1:
            y_score = y_score.reshape((-1, 1))
        n_classes = y_score.shape[1]
        score = np.zeros((n_classes,))
        for c in range(n_classes):
            y_true_c = y_true.take([c], axis=1).ravel()
            y_score_c = y_score.take([c], axis=1).ravel()
            score[c] = binary_metric(y_true_c, y_score_c)
        return np.average(score)

    return _average_binary_score(_binary_roc_aupr_score, y_true, y_score, average)


def drawing(d_result, contrast_list, info_list):
    column = []
    for i in contrast_list:
        column.append(i)
    df = pd.DataFrame(columns=column)
    if info_list[-1] == 'aupr':
        for i in contrast_list:
```

```
            df[i] = d_result[i][:, 1]
    else:
        for i in contrast_list:
            df[i] = d_result[i][:, 2]
    df = df.astype('float')
    color = dict(boxes='DarkGreen', whiskers='DarkOrange', medians='DarkBlue', caps='Gray')
    df.plot.box(ylim=[0, 1.0], grid=True, color=color)
    return 0


def save_result(feature_name, result_type, clf_type, result):
    with open(feature_name + '_' + result_type + '_' + clf_type+ '.csv', "w", newline='') as csvfile:
        writer = csv.writer(csvfile)
        for i in result:
            writer.writerow(i)
    return 0

def main(args):
    seed = 0
    CV = 5
    interaction_num = 10
    conn = sqlite3.connect("event.db")
    df_drug = pd.read_sql('select * from drug;', conn)
    df_event = pd.read_sql('select * from event_number;', conn)
    df_interaction = pd.read_sql('select * from event;', conn)

    feature_list = args['featureList']
    featureName="+".join(feature_list)
    clf_list = args['classifier']
    for feature in feature_list:
        set_name = feature + '+'
    set_name = set_name[:-1]
    result_all = {}
    result_eve = {}
    all_matrix = []
    drugList=[]
    for line in open("DrugList.txt",'r'):
        drugList.append(line.split()[0])
    if args['NLPProcess']=="read":
        extraction = pd.read_sql('select * from extraction;', conn)
        mechanism = extraction['mechanism']
        action = extraction['action']
        drugA = extraction['drugA']
        drugB = extraction['drugB']
    else:
        mechanism,action,drugA,drugB=NLPProcess(drugList,df_interaction)

    for feature in feature_list:
        print(feature)
        new_feature, new_label, event_num = prepare(df_drug, [feature], vector_size, mechanism,action,drugA,drugB)
        all_matrix.append(new_feature)

    start = time.clock()
```

```
    for clf in clf_list:
        print(clf)
        all_result, each_result = cross_validation(all_matrix, new_label, clf, event_num, seed, CV,
                                                   set_name)
        # ========================================================================
        #     save_result('all_nosim','all',clf,all_result)
        #     save_result('all_nosim','eve',clf,each_result)
        # ========================================================================
        save_result(featureName, 'all', clf, all_result)
        save_result(featureName, 'each', clf, each_result)
        result_all[clf] = all_result
        result_eve[clf] = each_result
    print("time used:", time.clock() - start)

if __name__ == "__main__":
    import argparse
    parser = argparse.ArgumentParser(formatter_class=argparse.ArgumentDefaultsHelpFormatter)
    parser.add_argument("-f","--featureList",default=["smile","target","enzyme"],help="features to use",nargs="+")
    parser.add_argument("-c","--classifier",choices=["DDIMDL","RF","KNN","LR"],default=["DDIMDL"],help="classifiers
to use",nargs="+")
    parser.add_argument("-p","--NLPProcess",choices=["read","process"],default="read",help="Read the NLP
extraction result directly or process the events again")
    args=vars(parser.parse_args())
    print(args)
    main(args)
```

ДОДАТОК Б ТОПОЛОГІЯ АРХІТЕКТУРИ ТРАНСФОРМЕРА

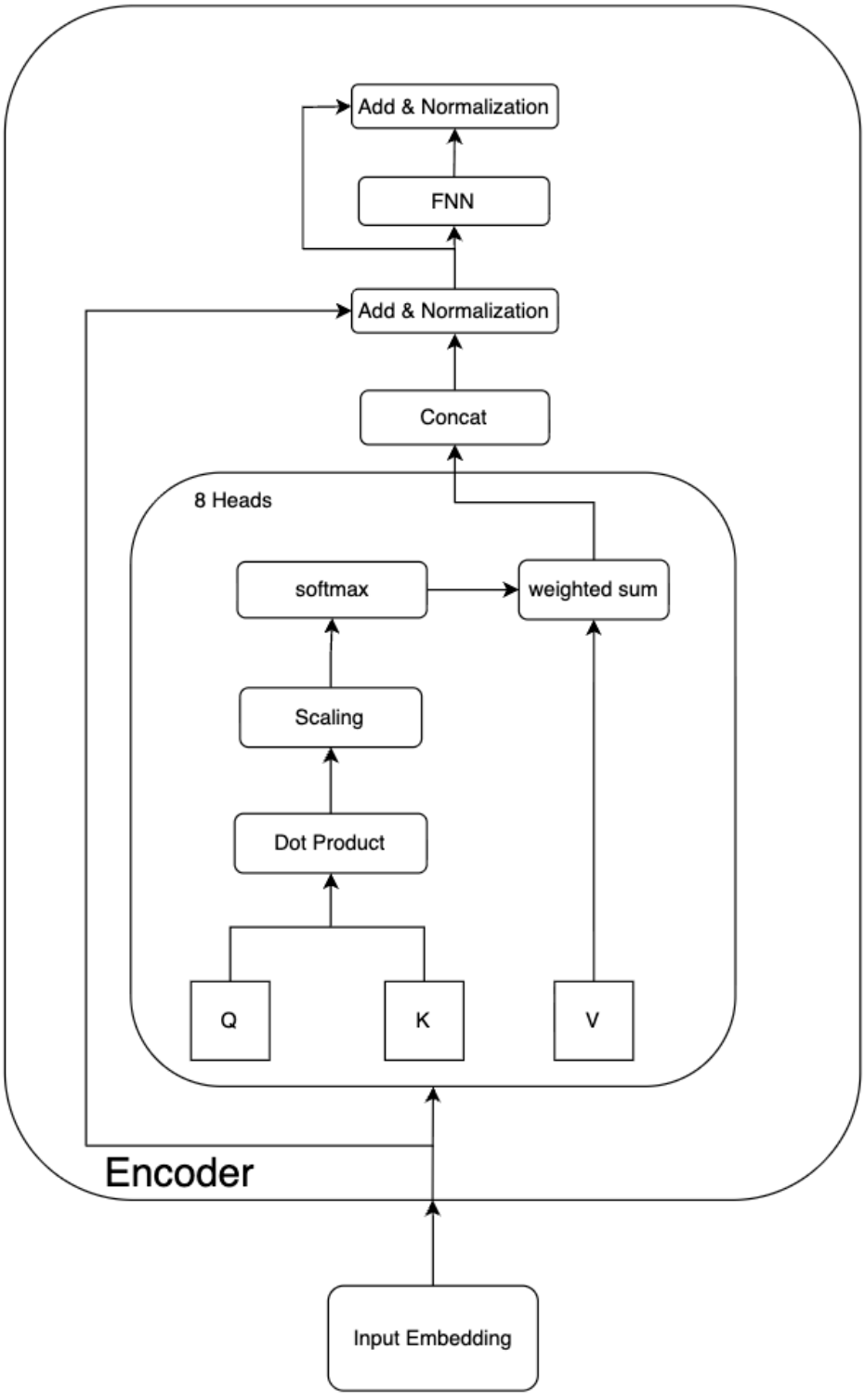

Рисунок Б.1 – Архітектура Transformer-Encoder блоку

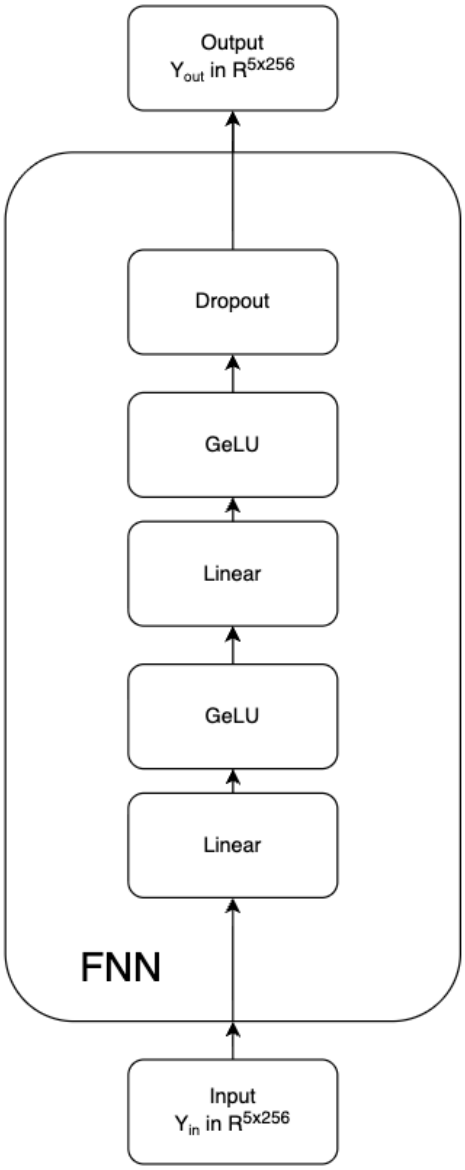

Рисунок Б.2 – Архітектура Feed-Forward мережі

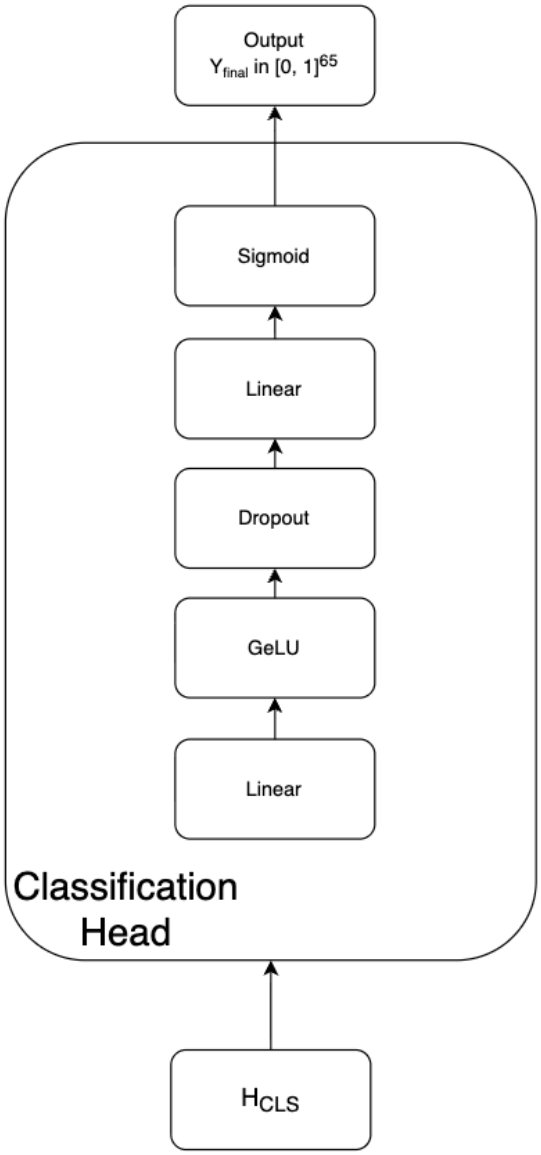

Рисунок Б.3 – Архітектура класифікатора