Drug Interaction Detection System

-An AI-Powered Clinical Safety Assistant

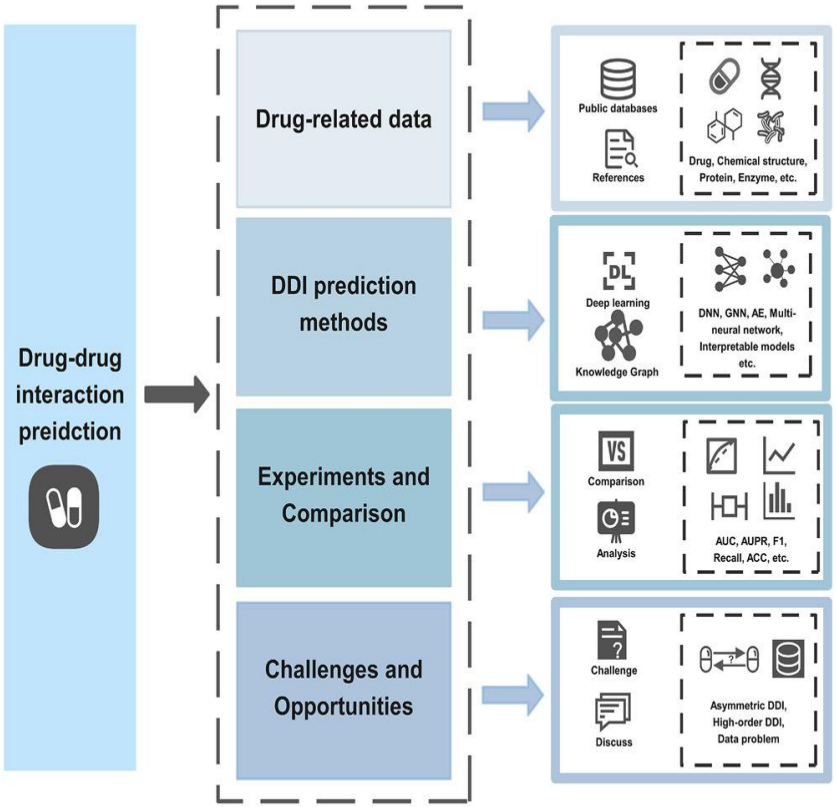
TITLE*: Drug Interaction Detection System — An AI-Powered Clinical Safety Assistant*
SUBTITLE: FastAPI + Streamlit with RxNorm, Medical NER, and IBM Watson NLU

TEAM: MEDICON
TEAM LEAD : CH.SAI GAYATHRI
TEAM MEMBERS : 1. PAGILLA SREE SAI VAGDEVI
2. MOGI NAVEENA
DATE : 09/03/2025

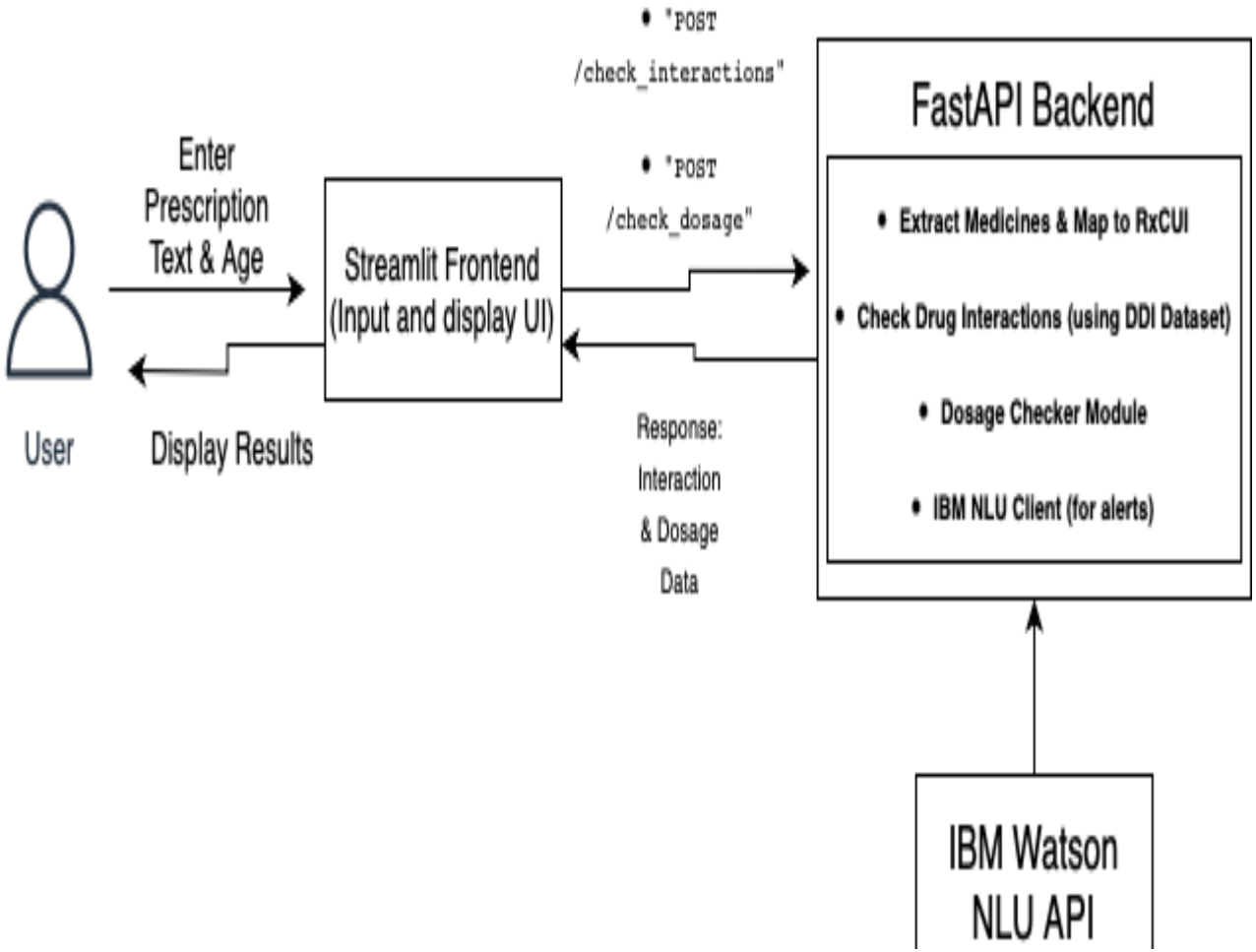
DESCRIPTION:

This project aims to analyze drug interactions, identify correct drug dosages, and provide safe alternative medication options based on age and drug details. It integrates multiple datasets and leverages advanced NLP models and APIs for accurate drug information extraction and interaction understanding. The system is built with a FastAPI backend and a Streamlit frontend for easy user interaction.

Data sources



MEDICAL PRESCRIPTION VERIFICATION LEV ERAGING



Problem statement and objectives

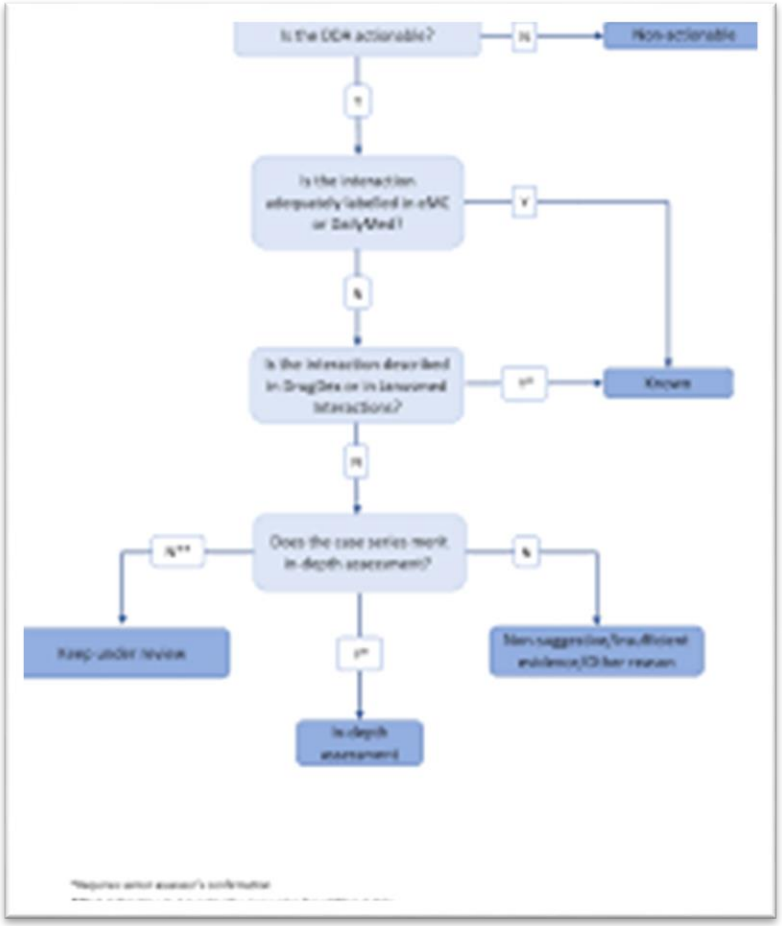
Problem Statement:

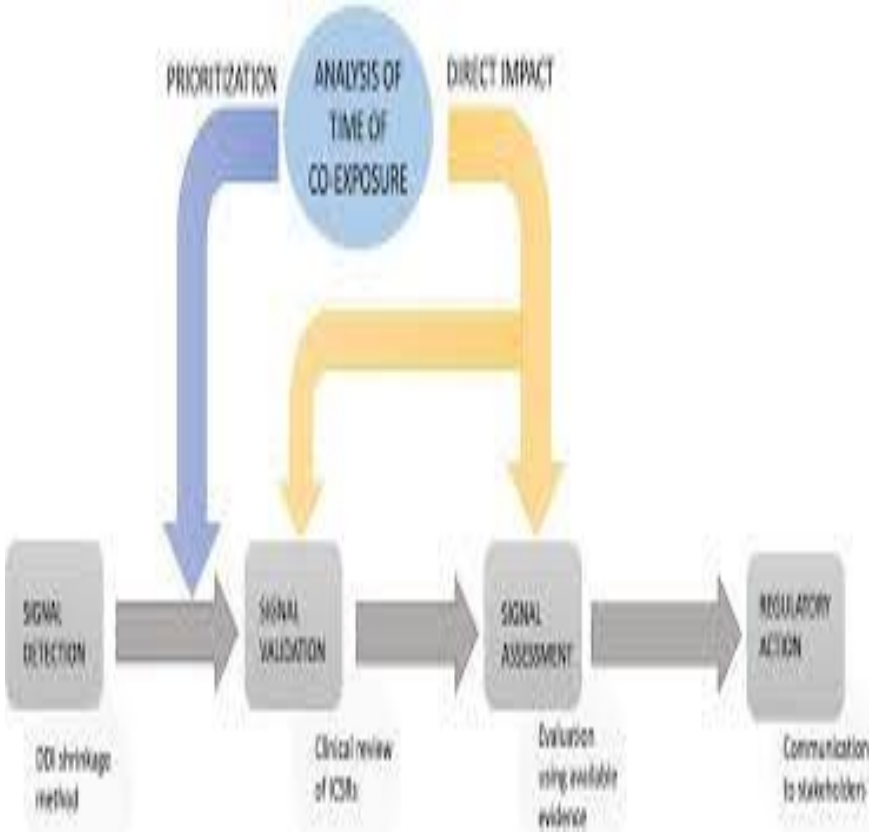
Clinicians and patients struggle with fragmented, non-standardized drug data.
Manual interaction checks are time-consuming and error-prone.
Explanations and dosage safety vary by age and context.

Objectives:

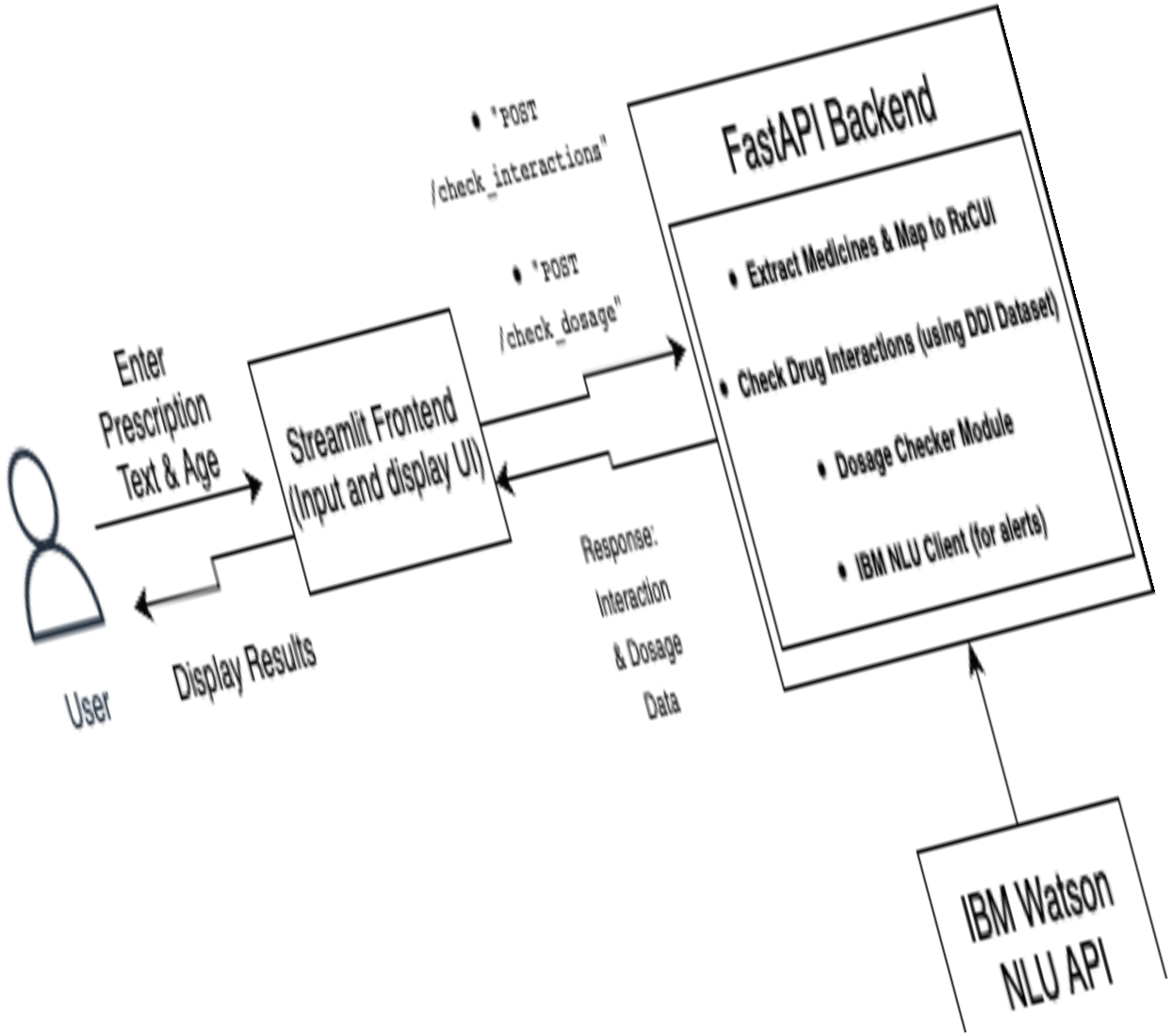
Automate drug extraction, standardization, and interaction detection.
Provide explainable, context-rich alerts.
Verify dosage and recommend safe alternatives.
Deliver a responsive UX for basic clinical workflows.

FLOW CHARTS


Is the CDA actionable?
Non-actionable
Is the interaction adequately labelled in eIAC or DailyMed?
Is the interaction described in DrugDex or in Lexicomp Interactions?
Known
Does the case series merit in-depth assessment?
In-depth assessment


PRIORITIZATION
ANALYSIS OF TIME OF CDI EXPOSURE
DIRECT IMPACT
SIGNAL DETECTION
SIGNAL VALIDATION
SIGNAL ASSESSMENT
REGULATORY ACTION
CDI shrinkage method
Clinical review of ICSRs
Evaluation using available evidence
Communication to stakeholders

WATSON AND HUGGING FACE MODELS



Pre-requisites

1. Python 3.8AI Medical Prescription Verification Leveraging IBM Watson and Hugging Face Models_Document.docx+
   - Installation guide: Python Official Site
   - Verify installation: python –version
2. FastAPI Framework
   - Official docs: FastAPI
   - Install with pip install fastapi
3. Streamlit for Frontend
   - Official docs: Streamlit
   - Install with pip install streamlit
4. Hugging Face API Key (for samant/medical-ner model)
   - Sign up at Hugging Face
   - Obtain API key to extract drug names and dosages
5. IBM Watson NLP AP
   - Sign up at IBM Cloud
   - Get API key and URL for NLP-based interaction understanding
6. RxNorm API Keys
   - Sign up for RxNav API
   - Used to map RxCUI to drug dosages and alternatives
7. Additional Python Libraries
   - requests, pandas, numpy, fastapi, streamlit
   - Install via pip install -r requirements.txt

Project Flow

Milestone 1: Data Acquisition and Integration

- Activity 1.1: Dataset Download
- Activity 1.2: Dataset mapping and Preparation

Milestone 2: NLP Model Integration for Drug Extraction and Interaction Understanding

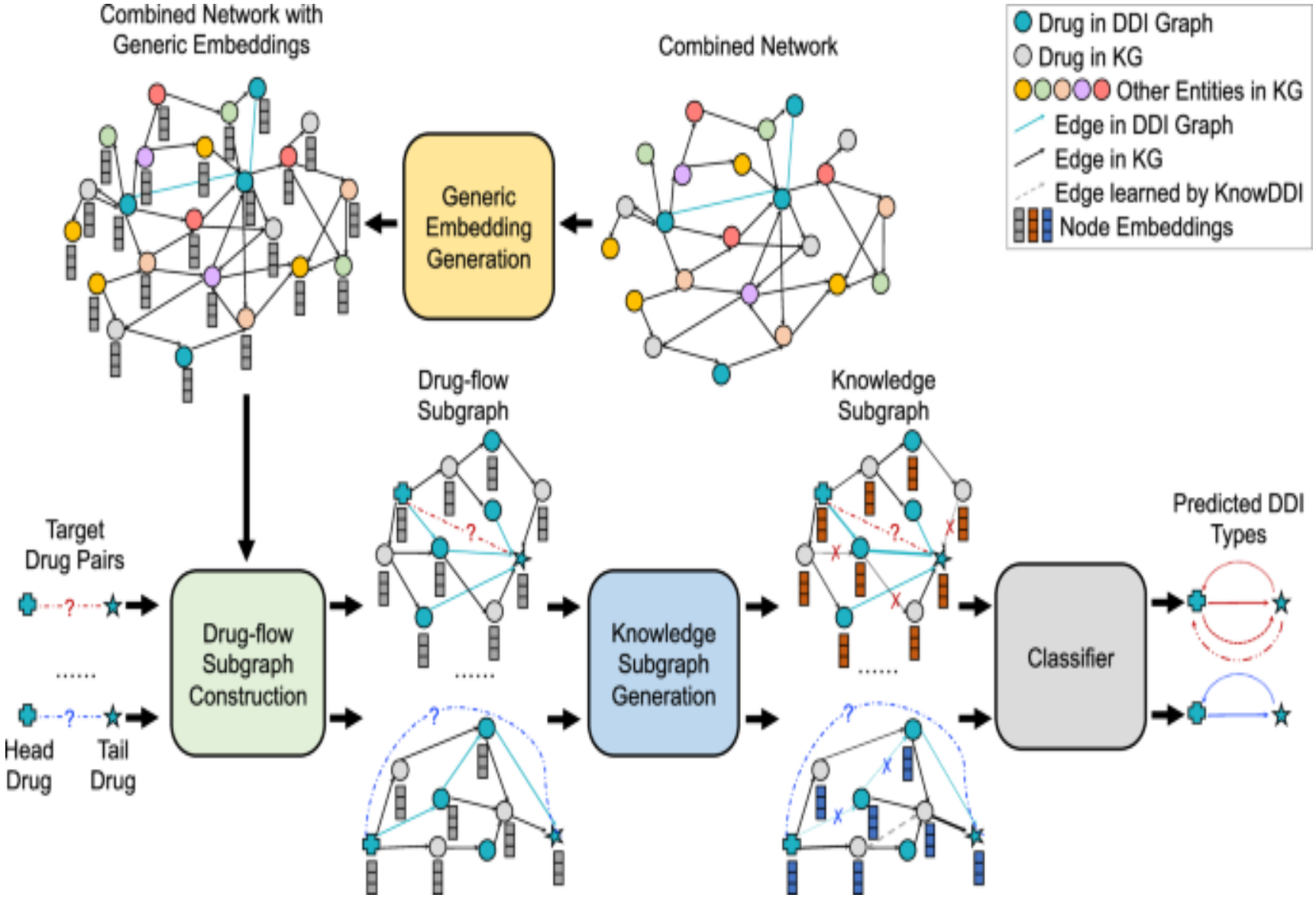
- Activity 2.1: Named Entity Recognition
- Activity 2.2: IBM Watson NLP for Interaction Context
- Activity 3.3: Integration of both the models

Milestone 3: Dosage Verification and Alternative Recommendation

- Activity 3.1: RxNorm API Usage
- Activity 3.2: Alternative Safe Drug Suggestions

Milestone 4: Backend and Frontend Development

- Activity 4.1: FastAPI Backend
- Activity 4.2: Streamlit Frontend


Combined Network with Generic Embeddings
Combined Network
Drug in DDI Graph
Drug in KG
Other Entities in KG
Edge in DDI Graph
Edge in KG
Edge learned by KnowDDI
Node Embeddings
Generic Embedding Generation
Drug-flow Subgraph
Knowledge Subgraph
Target Drug Pairs
Drug-flow Subgraph Construction
Knowledge Subgraph Generation
Classifier
Predicted DDI Types
Head Drug
Tail Drug

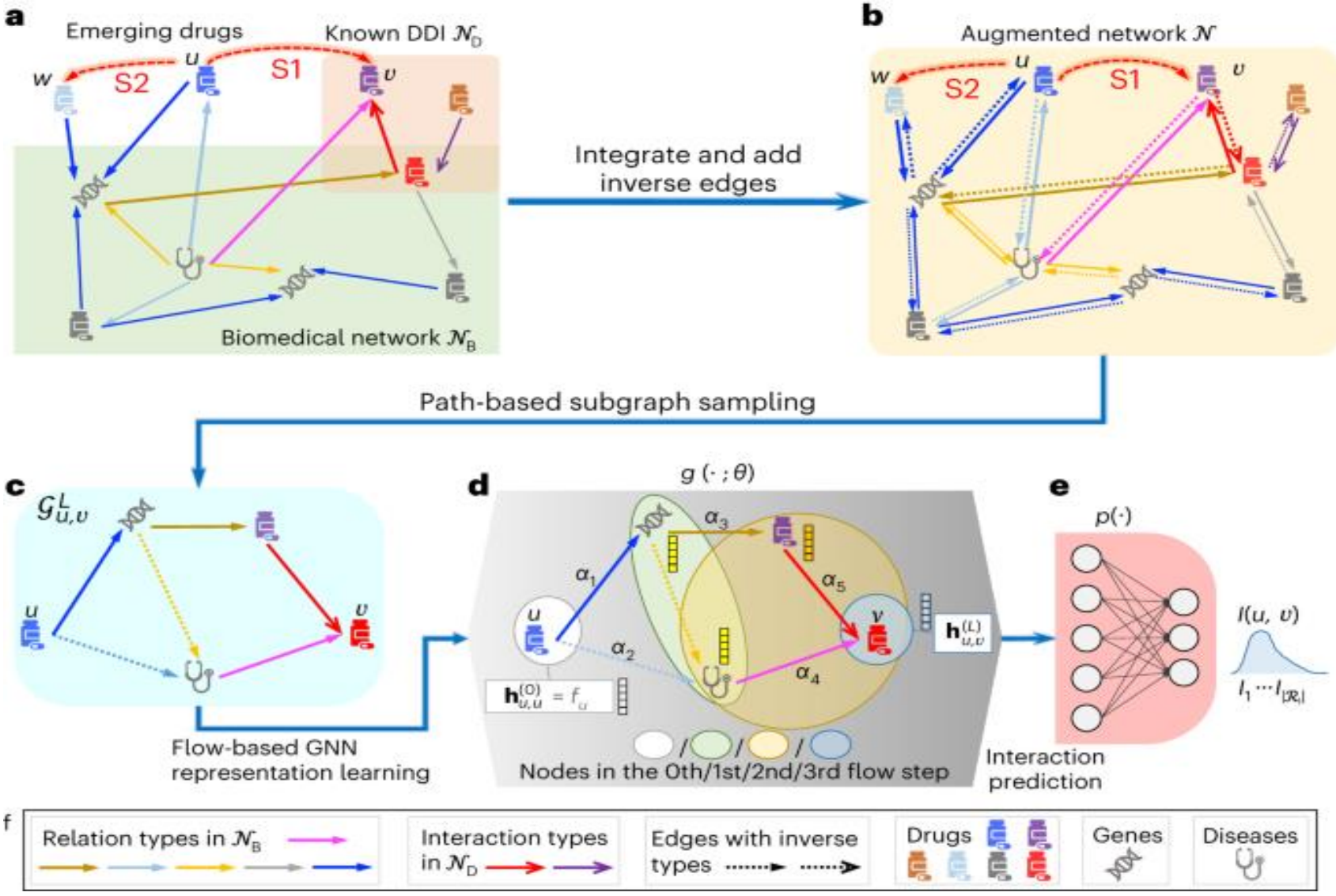
NLP pipeline: drug and dosage extraction

- Pipeline:

  Text cleaning → NER inference → merge fragmented tokens → remove noise → final list of drugs + optional dosages.

Capabilities:

Detects medications, strengths, units, and forms from unstructured text.


a
Emerging drugs
Known DDI $\mathcal{N}_D$
w
u
v
S2
S1
Biomedical network $\mathcal{N}_B$
Integrate and add inverse edges
b
Augmented network $\mathcal{N}$
Path-based subgraph sampling
c
$\mathcal{G}^L_{u,v}$
Flow-based GNN representation learning
d
$g(\cdot;\theta)$
α_1
α_2
α_3
α_4
α_5
$\mathbf{h}^{(0)}_{u,u} = f_u$
$\mathbf{h}^{(L)}_{u,v}$
Nodes in the 0th/1st/2nd/3rd flow step
e
$p(\cdot)$
$l(u, v)$
$l_1 \cdots l_{|\mathcal{R}|}$
Interaction prediction
f
Relation types in $\mathcal{N}_B$
Interaction types in $\mathcal{N}_D$
Edges with inverse types
Drugs
Genes
Diseases

Under the folder name, **AI_Prescription_Verifier**

This milestone focused on collecting and preparing high-quality datasets essential for accurate drug interaction analysis. We sourced raw interaction data from Kaggle and standardized drug details using the RXNORM dataset from the U.S. National Library of Medicine. By mapping and merging these datasets through RxCUIs, we created a comprehensive and scientifically consistent foundation to power subsequent analysis and model development. The first step involved acquiring reliable datasets related to drug interactions and drug information.Drug Interaction Dataset from KaggleWe downloaded the dataset from Kaggle, which contains pairs of drugs and their known interactions. This dataset provides raw data about which drugs interact with each other, but it mainly uses commercial or common drug names without standardized scientific identifiers.

Dosage verification and safe alternatives

Dosage check:

Cross-reference age and drug strength/form with RxNorm SCD properties.
Flag under/over-dosing relative to age.

Safe alternatives:

Resolve active ingredient (IN) → get SBD/SCD options.
Suggest therapeutically equivalent brands/generics.

Output:

“Recommended: 250–500 mg every 6–8 hours (age-based).
Consider [Alternative A/B] if interaction present.”

Performance, testing, and challenges

Performance:

Pre-indexed lookups for DDI pairs.
Caching of RxNorm and NER responses.

Testing:

Unit tests for mapping, API contracts, and edge input.
Functional tests through Swagger and Streamlit UI.

Challenges:

Noisy medical text, abbreviations, misspellings.
Third-party API latency and resilience.
Clear UI for multi-drug interactions.

Mitigations:

Normalization rules, retries/timeouts, pagination, compact tables.

THANK YOU…!!!!