# Survey on Using LLM Agents for Medical Data Cleaning

Siqi Yao

September 16, 2025

## Contents



## 1 Introduction

Data cleaning is a crucial yet tedious task, which has led to the pursuit of autonomous solutions, with LLM agents emerging as a promising approach. For instance, [BDH25] provides an in-depth exploration of how LLM agents can be used to automate the entire data cleaning process. The study evaluates this method across three Kaggle datasets—Titanic, Meat Consumptions, and Hotel Bookings—to assess whether data cleaning through agents alone can enhance the performance of machine learning models.

I searched for articles specifically focusing on the use of agents for cleaning medical or healthcare data, but could not find any. However, I did come across three relevant papers. [WZW+24] and [SVG+25] both discuss the construction of agents for end-to-end analysis of healthcare data, with data cleaning being just one step in the process. Meanwhile, [SPLF25] explores how LLM agents can be utilized for data harmonization, which involves merging data from different sources, using healthcare data as an example. Next, I will provide a brief overview of the background and key content of each paper, along with their approaches to data cleaning or harmonization.

# 2 *EHRFlow: A Large Language Model-Driven Iterative Multi-Agent Electronic Health Record Data Analysis Workflow*

## 2.1 Background

Electronic Health Records (EHRs) play a crucial role in healthcare, and there is a growing need for agents that can assist physicians in analyzing EHRs. The challenge lies in the need for these agents to address the specific, customized requirements and expertise of physicians while ensuring precise and reliable execution. Additionally, privacy concerns in healthcare data must be carefully managed, ensuring that no patient information is uploaded when utilizing online APIs for LLMs.

## 2.2 Key Content

[WZW+24] introduces the **EHRFlow**, an iterative multi-agent system supported by LLMs, designed to serve for physicians' diverse EHR analysis requirements in their clinical workflows. As shown in Figure 1, EHRFlow consists of four main agents: PlanAgent, ToolAgent, CodeAgent and ReviewAgent. A dual-loop mechanism is utilized. The outer loop formalize complex tasks into multiple coarse-grained plans, and each plan is further split into lower-level fine-grained executable tasks by the PlanAgent one by one. The inner loop refines and executes the task using tools retrieved from the **healthcare tool bank**, leveraging ToolAgent's reasoning ability for choosing required tool APIs and CodeAgent's code generation and execution capabilities. After finishing the task, PlanAgent continues to reason for the next task. Meanwhile, the ReviewAgent examines the execution results and provides reflection and feedback to the PlanAgent and ToolAgent. Specific examples can be found in Section 4 of the paper.

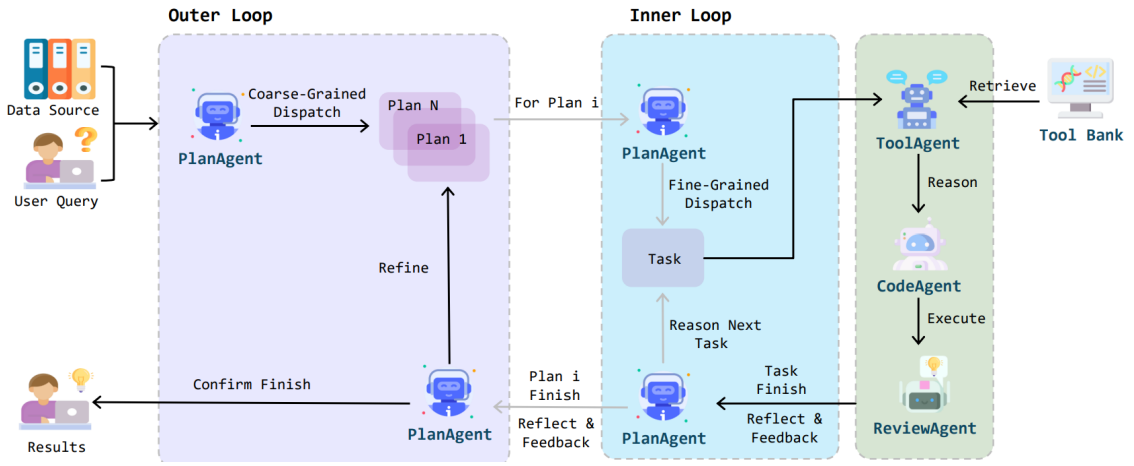


Figure 1: Framework of EHRFlow.

## 2.3 Approach to Data Cleaning

A series of steps including data cleaning, transformation, integration and advanced analysis are necessary to accurately extract and present the key information in EHR data. However, existing large language models do not provide sufficient support for the complex predictive analysis required.

To address this challenge, the paper adopted a functional decoupling strategy, separating the key functions of EHR data analysis from the system. They designed the **EHRFlowAPI**, a set of customized interfaces tailored to the specific needs and processes of medical data processing and analysis. The modular design of EHRFlowAPI allows each module to focus on specific tasks within the data processing and analysis workflow, covering the entire process from data preprocessing, feature extraction, model training and prediction, to data analysis interpretation and result

visualization. Through EHRFlowAPI, the role of ToolAgent shifts from generating complete analysis and prediction code to invoking these interfaces.

**Although the paper claims that the code for the tool banks and EHRFlow has been publicly released, I was unable to find the corresponding link, and the paper does not provide further details about the tools used. Perhaps we can ask the authors if needed.**

# 3 *Agentic AI framework for End-to-End Medical Data Inference*

## 3.1 Background

Although AI has the potential to transform clinical workflows, deploying machine learning models from raw clinical data remains fragmented, manual and costly. Additionally, the application of clinical AI faces further challenges related to privacy and data heterogeneity. As mentioned in previous sections, a promising strategy to address these challenges is to formalize AI systems as collections of autonomous, modular "agents", each assigned specific roles and goals. The tasks are then distributed among the agents to achieve more efficient and effective outcomes.

## 3.2 Key Content

[SVG+25] introduces an **Agentic AI** framework that modularly orchestrates clinical workflows, including data ingestion, anonymization, model training, inference, and explanation, through a network of domain-specialized agents. Together, these agents form an end-to-end system capable of autonomously translating raw, multimodal clinical data into privacy-preserving, interpretable predictions.

The complete architecture with all the agents interacting with each other and the data can be seen in Figure 2. The **Ingestion Identifier Agent** serves as the initial component of the pipeline, responsible for classifying uploaded files into recognized data formats to enable subsequent processing. The **Data Anonymizer Agent** ensures data privacy by performing automated detection and masking of personally identifiable information (PII). The **Feature Extraction Agent** performs automated, modality-specific feature identification, which is crucial for understanding the data. For structured data, the agent uses column names (also referred to as "headers") as proxies for feature descriptors, and are treated as candidate variables for subsequent tasks. For unstructured image data, the framework integrates MedGemma, a medical vision–language model developed by Google [SKJ+25]. A random anonymized image from the previous step is passed through a multi-stage classification pipeline to return image data-specific "headers": Modality and Disease Type. The **Model-Data Matcher Agent** connects raw data ingestion and model deployment by selecting the most appropriate model from a curated repository based on semantic alignment between input features and model requirements. The **Preprocessing Recommender Agent** and **Preprocessing Implementor Agent** will be introduced in Section 3.3. The final stage of the pipeline is handled by the **Model Inference Agent**, which is responsible for applying trained machine learning models to the processed input data and generating interpretable outputs. Figure 6 of the paper demonstrates a concrete example.

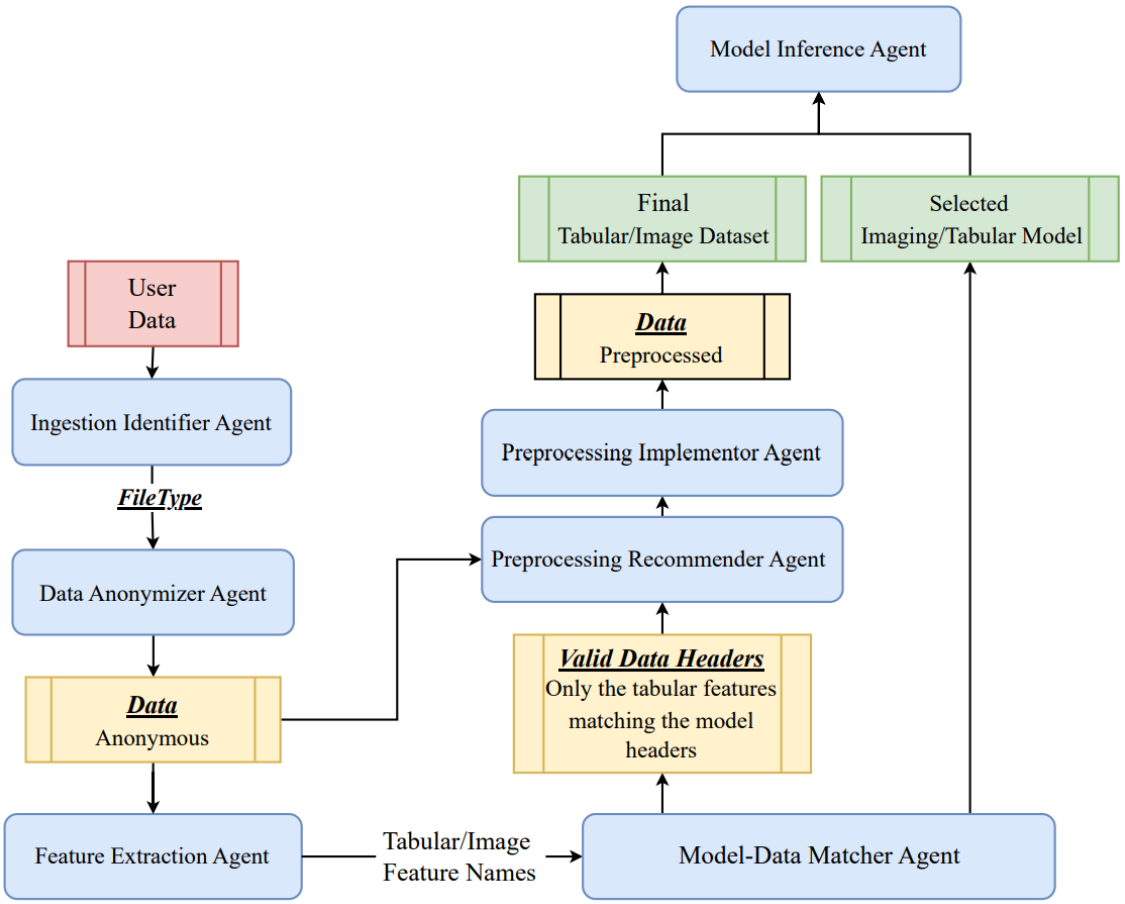


Figure 2: Complete architecture of Agentic AI.

### 3.3 Approach to Data Cleaning

The **Preprocessing Recommender Agent** autonomously suggests optimal preprocessing strategies that suits both the structure of the uploaded dataset and the specific requirements of the selected machine learning model, while supporting both user-guided and fully automated modes.

For tabular data, the agent first extracts metadata for each header, capturing attributes such as column name, data type, number of null values etc. Based on this metadata, it infers the column type, classifying each column as "Binary", "Categorical", "Numerical", or "Textual" based on a heuristic rule. This labeling guides the selection of preprocessing steps for each header.

In the case of image data, preprocessing is not manually configured by the user. Instead, the system employs a model-specific preprocessing pipeline that is tightly coupled with the selected image model. In other words, preprocessing routines are not generic but are instead co-trained with the model. These pipelines may include learned resizing strategies, custom normalization schemes or image tokenization methods optimized for the model's attention mechanism and feature extraction layers.

Right after the Preprocessing Recommender Agent, the **Preprocessing Implementor Agent** applies the selected preprocessing steps to the dataset. It takes the anonymized, feature-matched data from the previous steps and executes each preprocessing step given by the preprocessing recommender.

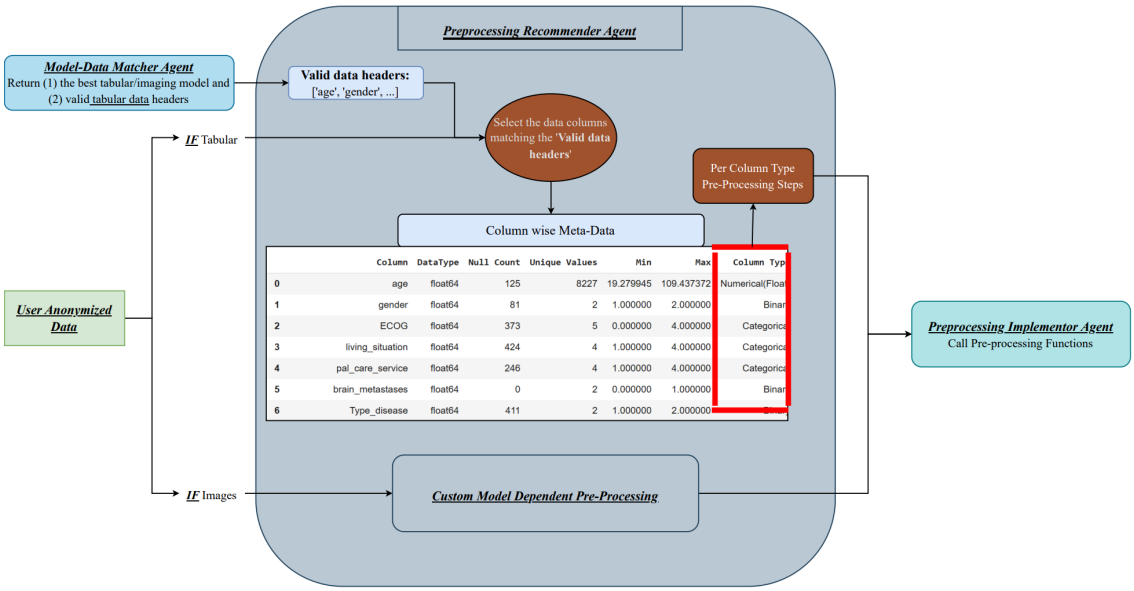


Figure 3: Preprocessing framework.

# 4 *Interactive Data Harmonization with LLM Agents: Opportunities and Challenges*

## 4.1 Background

Data harmonization refers to the practice of combining datasets from different sources to maximize their compatibility. An example is shown in Figure 4, where we aim to combine samples from two patient cohorts collected independently in two studies [DKG+23, DKZ+20]. We can see that the naming standards for the same quantity differ significantly between datasets. Thus, we wish to build a data harmonization agent that can produce a data processing pipeline that takes user data as input and outputs a harmonized table that satisfies the user requirements.

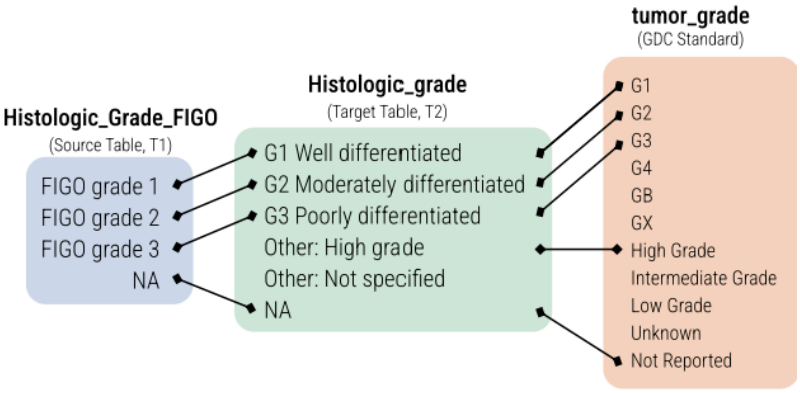


Figure 4: Domain of attributes in different data sources.

## 4.2 Approach to Data Harmonization

[SPLF25] propose using LLM-based agents to facilitate the interactive construction of harmonization pipelines through natural language and visual interfaces. The approach consists of three main components: harmonization primitives, harmonization agents and human-agent interaction, as shown in Figure 5.

The bottom of Figure 5 shows the **Harmonization Primitives** (**Data Integration Primitives**), which contains algorithms from bdi-kit, a Python library designed by the authors. The library provides state-of-the-art tools of data harmonization (with a focus on biomedical data) and APIs for these tools. Moreover, the library is specifically designed with composability in mind, enabling the seamless combination of different primitives to construct a robust data harmonization pipeline.

In the center of Figure 5 we find the **Data Harmonization Agent** (**LLM Agent**), which autonomously interacts with its environment—comprising data, primitives, users etc.—to execute harmonization tasks. It can function at different levels of autonomy, ranging from providing minimal user assistance to fully independent harmonization.

The last key aspect of the architecture is the **Human-Agent Interaction**. The authors show that the current prototype is proficient in text-based conversational interactions, and they also plan to equip the system with graphical user interfaces in the future.

Specific example of the whole procedure can be found in Section 4 of the paper.

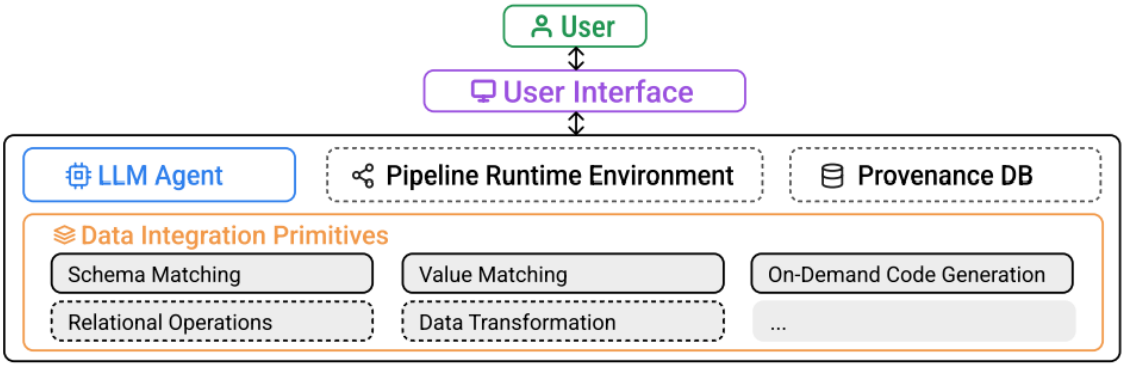


Figure 5: Components of an interactive agentic data harmonization system. Solid lines represent components implemented in Harmonia while dashed lines represent components not yet implemented.

## References


[BDH25] T. Bendinelli, A. Dox, and C. Holz. Exploring llm agents for cleaning tabular machine learning datasets. *arXiv preprint arXiv:2503.06664*, 2025.

[DKG+23] Y. Dou, L. Katsnelson, M. A. Gritsenko, Y. Hu, B. Reva, R. Hong, Y.-T. Wang, I. Kolodziejczak, R. J.-H. Lu, C.-F. Tsai, et al. Proteogenomic insights suggest druggable pathways in endometrial carcinoma. *Cancer Cell*, 41(9):1586–1605, 2023.

[DKZ+20] Y. Dou, E. A. Kawaler, D. C. Zhou, M. A. Gritsenko, C. Huang, L. Blumenberg, A. Karpova, V. A. Petyuk, S. R. Savage, S. Satpathy, et al. Proteogenomic characterization of endometrial carcinoma. *Cell*, 180(4):729–748, 2020.

[SKJ+25] A. Sellergren, S. Kazemzadeh, T. Jaroensri, A. Kiraly, M. Traverse, T. Kohlberger, S. Xu, F. Jamil, C. Hughes, C. Lau, et al. Medgemma technical report. *arXiv preprint arXiv:2507.05201*, 2025.

[SPLF25] A. Santos, E. H. Pena, R. Lopez, and J. Freire. Interactive data harmonization with llm agents: Opportunities and challenges. *arXiv preprint arXiv:2502.07132*, 2025.

[SVG+25] S. R. Shimgekar, S. Vassef, A. Goyal, N. Kumar, and K. Saha. Agentic ai framework for end-to-end medical data inference. *arXiv preprint arXiv:2507.18115*, 2025.

[WZW+24] H. Wu, Y. Zhu, Z. Wang, X. Zheng, L. Wang, W. Tang, Y. Wang, C. Pan, E. M. Harrison, J. Gao, et al. Ehrflow: A large language model-driven iterative multi-agent electronic health record data analysis workflow. In *KDD'24 Workshop: Artificial Intelligence and Data Science for Healthcare: Bridging Data-Centric AI and People-Centric Healthcare*, 2024.