

一般口演

一般口演5 医療支援

2022年11月18日(金) 13:10 ～ 14:55 H会場 (206会議室)

[2-H-2-01] 臨床現場への医用 AI導入のための医療情報・AI連携システムの開発

*中津井 雅彦¹、平野 靖²、安部 武志¹、濱田 和希³、幸田 恭治⁴、西中 芳幸⁵、石田 博²、浅井 義之^{3,1}（1. 山口大学大学院医学系研究科・医学部附属病院AIシステム医学・医療研究教育センター, 2. 山口大学医学部附属病院医療情報部, 3. 山口大学大学院医学系研究科システムバイオインフォマティクス講座, 4. 山口大学医学部附属病院薬剤部, 5. インタセクト・コミュニケーションズ株式会社）

*Masahiko Nakatsui¹, Yasushi Hirano², Takeshi Abe¹, Kazuki Hamada³, Kyoji Kouda⁴, Yoshiyuki Nishinaka⁵, Haku Ishida², Yoshiyuki Asai^{3,1}（1. 山口大学大学院医学系研究科・医学部附属病院AIシステム医学・医療研究教育センター, 2. 山口大学医学部附属病院医療情報部, 3. 山口大学大学院医学系研究科システムバイオインフォマティクス講座, 4. 山口大学医学部附属病院薬剤部, 5. インタセクト・コミュニケーションズ株式会社）

キーワード：Artificial Intelligence, Clinical Decision Support System, Integrative Framework

人工知能技術の飛躍的な発展や、膨大な医療情報の蓄積を背景として、医療の高精度化や効率化を目指し、様々な医用 AIの開発が進められている。医用 AIを臨床現場で活用していくためには、医用 AIと、電子カルテシステムや臨床意思決定支援システム（Clinical Decision Support System, CDSS）をはじめとする医療情報システムとの連携が不可欠である。また、臨床現場の多様なニーズやユースケースによって、予測に使用するデータの種別や実行のタイミングなどが異なる医用 AIが想定され、医用 AIの臨床現場への迅速な展開には、様々な形式の医用 AIを統一的な手順で臨床現場へと導入できることも重要である。そこで本研究では、多様な医療データを活用し、医用 AIを適切なタイミングで実行し、その予測結果を臨床現場へと提示する医用 AI・医療情報の汎用的な連携システムを構築する。

本連携システムでは、医用 AIの種類ごとに CDSSにおいてあらかじめ設定した判断ロジックに基づき、任意の医用 AIが起動される。判断ロジックとしては、特定のイベントの発生や異常検査値の検出、または日次などのバッチ処理を設定できる。医用 AIは、医療情報システムの二次利用システムから SQLを用いて患者データを取得し、予測を実行する。予測結果は CDSSへと返却され、その結果が医療情報システムの画面上に表示される。医用 AIは、他の医療情報システムから独立した専用仮想サーバ上で実行する。また、コンテナ仮想化技術である Dockerを用いることで、他の医用 AIの追加や改修による動作環境変化の影響を受けない。

本発表では、有害反応の疑いがある場合に過去の副作用発生情報に基づきその原因薬を予測する AIを題材として、連携システムの実装、および山口大学医学部附属病院における医用 AIの臨床現場への導入に向けた試行と、将来に向けた展望について報告する。

臨床現場への医用 AI 導入のための医療情報・AI 連携システムの開発

中津井雅彦^{*1}、平野靖^{*2}、安部武志^{*1}、濱田和希^{*3}、幸田恭治^{*4}、
西中芳幸^{*5}、石田博^{*6}、浅井義之^{*1,3}

^{*1} 山口大学大学院医学系研究科・医学部附属病院 AI システム医学・医療研究教育センター、

^{*2} 山口大学医学部附属病院 医療情報部、

^{*3} 山口大学大学院医学系研究科 システムバイオインフォマティクス講座、^{*4} 山口大学医学部附属病院 薬剤部、

^{*5} インタセクト・コミュニケーションズ株式会社、^{*6} 山口大学大学院医学系研究科 医療情報判断学講座

Development of integrative framework for medical information system and medical AI

Masahiko Nakatsui^{*1}, Yasushi Hirano^{*2}, Takeshi Abe^{*1}, Kazuki Hamada^{*3}, Kyoji Kouda^{*4},
Yoshiyuki Nishinaka^{*5}, Haku Ishida^{*6}, Yoshiyuki Asai^{*1,3}

^{*1} AI Systems Medicine Research and Training Center, Yamaguchi University,

^{*2} Department of Medical Informatics and Decision Sciences, Yamaguchi University Hospital,

^{*3} Department of Systems Bioinformatics, Yamaguchi University Graduate School of Medicine,

^{*4} Pharmacy Department, Yamaguchi University Hospital,

^{*5} Intasect Communications, Inc.,

^{*6} Department of Medical Informatics and Decision Sciences, Yamaguchi University Graduate School of Medicine

Abstract

In order to utilize medical AI in clinical practice, it is essential to integrate medical AI with medical information systems, including electronic medical record systems and clinical decision support systems (CDSS). In this study, we developed integrative framework for medical AI/medical information system that utilizes a variety of medical data, executes medical AI at the appropriate timing, and notices the predicted results to the clinical site. In the developed system, medical AI is activated by the occurrence of specific medical events, such as detection of abnormal clinical test results, or periodic processing through batch processing. The medical AI acquires data from the DWH, which is a secondary use database for medical information. Prediction results are displayed on the screen of the medical information system. To ensure reproducibility of prediction results, the system is implemented using Docker technology.

In this presentation, we will report the implementation of the integrative framework, introduction trial of AI system for estimating culprit drugs for adverse drug reactions based on Bayesian inference into clinical practice at Yamaguchi University Hospital, and prospects for the future work.

Keywords: Artificial Intelligence, Clinical Decision Support System, Integrative Framework

1. 緒論

人工知能技術の飛躍的な発展や、膨大な医療情報の蓄積を背景として、医療の高精度化や効率化を目指し、様々な医用 AI の開発が進められている。山口大学においても、2018 年に AI システム医学・医療研究教育センター (AI Systems Medicine Research and Training Center, AISMEC) を設立し、AI 技術の基礎医学および臨床医学 (医療) への実践的な導入を目指している。多岐にわたる臨床現場のニーズに対応する医用 AI を臨床現場で活用していくためには、医用 AI と、多種多様な患者データを蓄積する電子カルテや診療科データベース、電子カルテシステムや臨床意思決定支援システム (Clinical Decision Support System, CDSS¹⁾) をはじめとする医療情報システムとの連携が不可欠である。また、臨床現場の多様なニーズやユースケースによって、予測に使用するデータの種別や実行のタイミングなどが異なる医用 AI が想定され、医用 AI の臨床現場への迅速な展開には、様々な形式の医用 AI を統一的な手順で臨床現場へと導入することも重要である。本研究では、多様な医療データを活用し、医用 AI を適切なタイミングで実行し、その予測結果を臨床現場へと提示する医用 AI・医療情報の汎用的な連携システムを

構築した。本稿では、開発した連携システムの概要について報告する。

2. 目的

様々な臨床ニーズに対応する医用 AI をスムーズに医療情報システムと連携させることにより、医用 AI 開発から臨床現場への展開までのリードタイムの短縮を目指す。医用 AI による診察のサポートが早期に実現できるようになり、医療の高度化及び効率化が期待できる。

3. 方法

検査値異常の検出などのイベント発生や定時のバッチ処理を起点として、医用 AI が自動実行され、予測結果が医療情報システムに提示されるシステムである。図 1 に、本システムの概要図を示す。本システムは、医用 AI 起動用プログラム、タスク管理 Key-Value Store、医用 AI を実行する仮想化コンテナで構成され、医療情報システムの CDSS、および電子カルテの二次利用データベースである DWH と連携して動作する。医用 AI の実行による計算機リソースの消費が医療情報システムへ悪影響を与えないよう、医用 AI を実行する計算機サーバは、医療情報システムから切り離された仮想サーバ

(Windows Server 2019 Standard Edition, VMware ESXi による仮想化)を用いる。

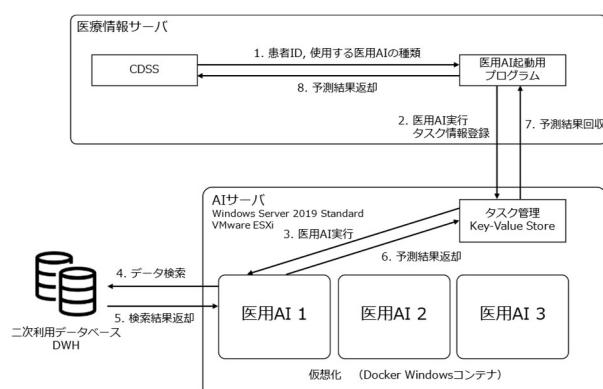


図 1 システム概要図

3.1 処理の流れ

CDSS に設定された判断ロジックに基づき医用 AI による予測が必要となった際に、以下の手順で処理が実行される。

1. 医用 AI による予測が必要な患者 ID、および起動する医用 AI の種類が、CDSS の動作するサーバ内に設置された医用 AI 起動用プログラムに伝達される。
2. 医用 AI 起動用プログラムが、タスク管理 Key-Value Store へ、予測に使用する医用 AI の種別を表す識別名および対象となる患者 ID を書き込む。
3. 自身と同じ識別名がタスク管理 Key-Value Store へ書き込まれた医用 AI プログラムが起動する。
4. 医用 AI プログラムは、DWH に対し、予測に必要なデータの検索を実施し、
5. 予測に必要なデータが医用 AI プログラムへ返却される。
6. 医用 AI プログラムによる予測が実行され、その結果がタスク管理 Key-Value Store へ書き込まれる。
7. 医用 AI 起動用プログラムは、タスク管理 Key-Value Store に書き込まれた予測結果を取得し、ファイル出力を行う。
8. CDSS が医用 AI 起動用プログラムにより出力されたファイルを読み取り、設定に従って医用 AI の実行状況や予測結果をユーザへ通知する。

3.2 医用 AI 起動用プログラム

医用 AI 起動用プログラムは、CDSS に設定された判断ロジックに基づき医用 AI による予測が必要となった際に、CDSS の外部アプリ実行機能を用いて呼び出されるプログラムであり、本連携システムの動作の起点である。入力ファイル名、出力ファイル名の二種類の引数を取り、起動する医用 AI の種類と入力ファイルの内容を、後述のタスク管理 Key-Value Store へと書き込む。また、医用 AI による予測が終了した際に、タスク管理 Key-Value Store から予測結果を取得し、出力ファイルへ出力する。CDSS は、出力ファイルに記載された予測結果をユーザへ通知する。

医用 AI には、種類を識別するための識別名が割り当てられており、CDSS は起動する医用 AI の識別名と同じ名前のディレクトリに入出力ファイルを設置し、医用 AI 起動用プログラムを起動する。医用 AI 起動用プログラムは、そのディレクトリ名を医用 AI の識別名として、入力ファイルの内容とともに、タスク管理 Key-Value Store へ書き込む。入力ファイルは、医

用 AI による予測の対象となる患者 ID を、各行に 1ID ずつ記載したテキストファイルである。医用 AI の種別によって必要なデータの種類、形式やサイズが異なるため、CDSS から本連携システムへ入力されるのは医用 AI の種類と患者 ID のリストのみであり、予測に必要な患者データについては、後述の医用 AI 実行プログラムが必要に応じて個別に取得する。

出力ファイルは CSV 形式で出力され、一行に一名の患者に対する予測結果を記載する。項目(カラム)は、患者 ID、医用 AI 実行のステータスコード(エラーコードも含む)、予測結果である。CDSS は、患者 ID ごとに、ステータスコードの値に従って、通知の有無や内容の選択を行う。

医用 AI 起動用プログラムの開発には、将来の移植性を考慮し、クロスプラットフォームでの開発が可能な Go 言語²⁾を用いた。

3.3 タスク管理 Key-Value Store

医用 AI 起動用プログラムと、医用 AI プログラムとの間で、タスク情報及び予測結果のやり取りを行うためのインメモリデータベースである。MSOpenTech によりリリースされた Redis on Windows 3.0.504³⁾を用いて構築した。AI サーバにおいて、Docker 技術⁴⁾による仮想化を用いて動作する。

3.4 医用 AI プログラム

医用 AI による予測を行う本体のプログラムである。また、予測に必要なデータを、医療情報の二次利用データベースとして整備された DWH より取得する。タスク管理、予測に必要なデータの取得、ログ出力、およびタスク管理 Key-Value Store との通信については、医用 AI の種類を問わず必要な機能であるため、共通のフレームワークとして準備し、様々な種類の医用 AI を容易に実装可能とした。

共通フレームワークは Python 言語 (3.9)⁵⁾を用いて実装した。タスクキュー管理には Celery 5.1.2⁶⁾を使用した。また、予測に必要なデータは、DWH に対し、JDBC⁷⁾を使用して SQL 文を実行することで取得できる。

医用 AI は、Docker 技術を活用し、AI サーバ上でさらに個別の仮想環境で動作することで、他の医用 AI の追加や改修に伴う動作環境(使用する外部プログラムのバージョンなど)が変わらず、AI の実行に影響しない設計とした。

4. 実装例

山口大学大学院医学系研究科において開発された、副作用原因薬予測 AI (Kazuki Hamada, et al., 論文投稿準備中) を本連携システムに実装し、2022 年 8 月 24 日より、山口大学医学部附属病院の医療情報システムとの連携テストを開始した。

4.1 副作用原因薬予測 AI

副作用原因薬予測 AI は、患者に副作用の疑いが発生した場合に、その患者に処方された薬剤に対して、副作用の原因薬である可能性を推定し、スコア付けすることを目的とする。独立行政法人医薬品医療機器総合機構 (PMDA) の提供する医薬品副作用データベース JADER⁸⁾へ報告された過去の副作用発生事例及び原因薬の情報から、ベイズ推定モデルを構築した。山口大学医学部附属病院での副作用報告 67 件、1,387 薬剤(原因薬 67, 併用薬 1,320)を用いてモデルの評価を行った結果、本モデルは ROC0.93 という高い性能を示した。本モデルの予測結果を医療従事者に提供することにより、副作用の原因薬を特定する一つの手がかりとなり、治療期間の短縮や診断制度の向上により、患者の不利益の軽減

が期待される。

4.2 連携システムへの組み込みと医療情報システムとの連携

以下の条件で、本連携システムへの組み込みと、医療情報システムとの連携を実施した。

本運用では、毎正時のバッチ処理とする。ただし、テスト運用は1日に1回の処理から開始し、CDSSシステムの負荷状況を確認しながら、処理の間隔を狭めていく。前回実行時以降からバッチ処理の開始時まで、医療情報システムにおける38項目の「疑いのある有害反応」に数値が入った患者IDを予測の対象とする。「疑いのある有害反応」の項目名とJADERで使用する副作用名(ICH国際医薬品用語集日本版 MedDRA/J⁹⁾の器官別大分類 Preferred Term, PT)とは異なるため、あらかじめ院内で作成した対応表によりMedDRA/JのPT名へ変換して使用する。また、医薬品名については、製品名ではなく、一般財団法人日本医療情報センター(JAPIC)により提供される医薬品名称データ¹⁰⁾を用い、一般名へ統一して取り扱う。患者への処方情報については、院内の医療情報の二次利用データベースから、過去180日間の「処方」および「注射」があった薬剤を取得する。

開発した共通フレームワークを用いて、副作用原因薬予測AIを本連携システムへ組み込み、Javaの実行環境(OpenJDK 17.0.1)、Python 3.9.7、JDBCドライバ(DWHベンダの提供)、Cライブラリおよびコンパイラ(マイクロソフト C++ビルドツール 2019)及び必要な外部Pythonライブラリとともに、Dockerイメージを作成した。医用AIを実行するAIサーバがWindows Server 2019 Standard Editionであるため、Dockerイメージの親イメージはWindows Server Core ltsc2019を用い、Windowsコンテナとして構築した。また、同様にWindowsコンテナとして、タスク管理用Key-Value Store(Redis on Windows 3.0.504)を動作させるためのDockerイメージを作成し、医用AI・タスク管理用Key-Value Storeの両DockerイメージをAIサーバへ設置し、起動した。

CDSSでは、医用AI起動用プログラムをCDSSが稼働するサーバへ設置し、CDSSの外部アプリとして登録した。また、前回の実行以降に副作用の疑いが発生した患者IDを取得し、医用AI起動用プログラムを起動し、予測結果を受け取りテスト運用の対象となるユーザへ通知するためのルール設定を行った。

4.3 結果

テスト運用において、2022年8月25日17時～2022年9月7日17時の期間において、副作用原因薬予測の対象となる患者は合計20名、一日平均約1.5名(平日約2名、土日祝0名)であり、CDSSによる検出、副作用原因薬推定システムによる予測、結果の指定ユーザへの通知が正常に実行された。

5. 考察

本連携システムは、計算機システムのハードウェア・ソフトウェアの変更や、他施設への展開も見据え、オープンソースソフトウェアの活用し、またクロスプラットフォームのプログラム開発言語を用いるなど、移植性およびクロスプラットフォームでの実行を念頭において開発してきた。また、外部ライブラリなどの実行環境が変化しても予測結果の再現性が担保できるよう、仮想化技術であるDocker技術も活用した。また、複数の種類の医用AIを組み込むことを想定し、共通化可能な機能は共通フレームワークとして実装するなど、汎用化を心が

けて開発を行ってきた。一方で、現時点での連携システムへの組み込みは一つの医用AIに限られており、複数の医用AIを組み込む際の知見をもとに、さらに連携システムの汎用化を進めていく必要がある。また、新規医用AIの連携システムへの組み込みは、その医用AIを含むDockerイメージを構築し、AIサーバ上で起動することにより可能であるが、Dockerイメージの構築についても、組み込むソフトウェアを厳選し、Dockerイメージのサイズを削減するなど、さらなる改良が必要である。

連携システムの汎用化、共通化、効率化を進める一方で、医療現場におけるAIに対するニーズは多様であり、医用AIを実行するための判断ロジック、予測結果を使用するデータの内容・期間、予測結果の内容、通知の内容、通知対象のユーザなど、ニーズに基づいて柔軟な設定が求められる部分もあり、その見極めが重要である。また、テスト運用などを通じてユーザの意見を汲み、改良を繰り返しながら、現場において有用かつ使いやすいシステムとして構築していくことが重要と思われる。

6. 結論

医療情報・医用AI連携システムを構築し、まずは副作用原因薬予測AIを組み込み、実際の医療情報システムを使用してテスト運用を開始した。今後は、テスト運用で得られた知見を踏まえ、本格運用へ向けた調整を実施する。

現時点での連携システムへの組み込みは副作用原因薬予測AIに限られるが、今後は本学医学系研究科・医学部附属病院で開発されている他の種類の医用AIについても連携システムへの組み込みを進め、その過程で得られた知見をもとに、新規医用AIの連携システムへの組み込み手順や、その際に使用する共通フレームワークのさらなる汎用化を目指す。

また、医療情報システムと医用AIとを連携させることの有用性の評価や、その方法についても引き続き検討を進める。

参考文献

- 1) Berner, Eta S., ed. Clinical Decision Support Systems. New York, NY: Springer, 2007.
- 2) Go [https://golang.org/doc/ (cited 2022-Sep-7)].
- 3) MSOpenTech Redis on Windows 3.0 Release Notes [https://github.com/microsoftarchive/redis/releases (cited 2022-Sep-7)].
- 4) Docker [https://www.docker.com/ (cited 2022-Sep-7)].
- 5) Python Release Python 3.9.7 [https://www.python.org/downloads/release/python-397/ (cited 2022-Sep-7)].
- 6) Celery – Distributed Task Queue [https://docs.celeryq.dev/en/stable/index.html (cited 2022-Sep-7)].
- 7) Java SE Technologies – Database [https://www.oracle.com/java/technologies/javase/javase-tech-database.html (cited 2022-Sep-7)].
- 8) 医薬品副作用データベース JADER (PMDA) [https://www.pmda.go.jp/safety/info-services/drugs/adr-info/suspected-adr/0003.html (cited 2022-Sep-7)].
- 9) MedDRA/J [https://www.jmo.pmrj.jp/ (cited 2022-Sep-7)].
- 10) 医薬品名称データ(JAPIC) [https://www.japic.or.jp/service/iyaku/iyaku_name.html (cited 2022-Sep-7)].
- 11) Windows Server Core By Microsoft [https://hub.docker.com/_/microsoft-windows-servercore (cited 2022-Sep-7)].