Computer Methods and Programs in Biomedicine 255 (2024) 108336

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: https://www.sciencedirect.com/journal/computer-methods-and-programs-in-biomedicine

# MoCab: A framework for the deployment of machine learning models across health information systems

Zhe-Ming Kuo [a], Kuan-Fu Chen [b,c,d], Yi-Ju Tseng [e,f,*]

[a] Department of Information Management, National Central University, Taoyuan, Taiwan
[b] College of Intelligent Computing, Chang Gung University, Taoyuan, Taiwan
[c] Medical Statistics Research Center, Chang Gung University, Taoyuan, Taiwan
[d] Department of Emergency Medicine, Chang Gung Memorial Hospital, Keelung, Taiwan
[e] Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan
[f] Computational Health Informatics Program, Boston Children's Hospital, Boston, MA, USA

ARTICLE INFO

*Keywords:*
Fast healthcare interoperability resources (FHIR)
Clinical decision support
Machine learning
SMART on FHIR

ABSTRACT

**Background and Objective:** Machine learning models are vital for enhancing healthcare services. However, integrating them into health information systems (HISs) introduces challenges beyond clinical decision making, such as interoperability and diverse electronic health records (EHR) formats. We proposed Model Cabinet Architecture (MoCab), a framework designed to leverage fast healthcare interoperability resources (FHIR) as the standard for data storage and retrieval when deploying machine learning models across various HISs, addressing the challenges highlighted by platforms such as EPOCH®, ePRISM®, KETOS, and others.
**Methods:** The MoCab architecture is designed to streamline predictive modeling in healthcare through a structured framework incorporating several specialized parts. The Data Service Center manages patient data retrieval from FHIR servers. These data are then processed by the Knowledge Model Center, where they are formatted and fed into predictive models. The Model Retraining Center is crucial in continuously updating these models to maintain accuracy in dynamic clinical environments. The framework further incorporates Clinical Decision Support (CDS) Hooks for issuing clinical alerts. It uses Substitutable Medical Apps Reusable Technologies (SMART) on FHIR to develop applications for displaying alerts, prediction results, and patient records.
**Results:** The MoCab framework was demonstrated using three types of predictive models: a scoring model (qCSI), a machine learning model (NSTI), and a deep learning model (SPC), applied to synthetic data that mimic a major EHR system. The implementations showed how MoCab integrates predictive models with health data for clinical decision support, utilizing CDS Hooks and SMART on FHIR for seamless HIS integration. The demonstration confirmed the practical utility of MoCab in supporting clinical decision making, validated by its application in various healthcare settings.
**Conclusions:** We demonstrate MoCab's potential in promoting the interoperability of machine learning models and enhancing its utility across various EHRs. Despite facing challenges like FHIR adoption, MoCab addresses key challenges in adapting machine learning models within healthcare settings, paving the way for further enhancements and broader adoption.

## 1. Introduction

Prediction models have become indispensable in modern healthcare [1] and other fields [2]. They play a crucial role in decision making, such as predicting consumer behavior [3], and forecasting weather patterns to assess the impacts of climate change [2]. In clinical settings, researchers extract information from diagnosis history, laboratory results, demographic information, and extensive clinical data [4] to build models that accurately predict patient outcomes and aid diagnostic processes [5]. Each of these applications relies on the ability to analyze large amounts of data to predict future states accurately.

The increasing adoption of predictive models in clinical practice has highlighted the need for a unified management system capable of streamlined workflows [6]. Integrating health information systems (HISs) with predictive models is essential to automate prediction

* Corresponding author at: Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan.
*E-mail address:* yjtseng@nycu.edu.tw (Y.-J. Tseng).

https://doi.org/10.1016/j.cmpb.2024.108336
Received 18 April 2024; Received in revised form 13 July 2024; Accepted 17 July 2024
Available online 20 July 2024
0169-2607/

processes. However, this integration is challenging due to the diverse electronic health record (EHR) formats in healthcare facilities [7]. The different formats of the EHR and the data entry requirements pose issues for clinicians, leading to new documentation modalities [8]. A previous study proposed an architecture that extracts information from various formats, including Fast Healthcare Interoperability Resources (FHIR), CDA R2, or natural language text documents, to improve healthcare care management [9]. Furthermore, the InteropEHRate project introduces protocols with FHIR standards to improve accessibility and utility across different HISs and EHR formats [10]. The variety in EHR systems extends beyond the data formats; it encompasses different terminologies, data standards, and system architectures. This diversity often results in fragmented health data ecosystems where crucial clinical information remains inaccessible, compounding the complexity of deploying predictive models across various EHR systems. Not only in the clinical domain, but machine learning is also inherently dependent on data, and format standardization is one of the key components to ensure interoperability in all applications [11]. These diversities require repeated adjustments to data retrieval procedures when transitioning between EHR systems and predictive models, imposing the complex task of applying a single model consistently across different HISs.

The precision of clinical decision support (CDS) systems is based on a regular update of predictive models with new data [12,13]. This is essential to counteract the performance degradation caused by outdated information [6], known as concept drift in machine learning, causing model deterioration [14]. The shift from traditional evidence-based medicine to "medicine-based evidence" underscores the importance of considering a broader data context surrounding patients [15,16]. In this context, continuous training of machine learning models is of vital importance [6,17,18], empowering these models to adapt to emerging trends and evolving patterns, thus ensuring precise predictions [19].

To address these challenges, we propose the Model Cabinet Architecture (MoCab), a centralized repository engineered to store and manage various predictive models, including scoring [20,21], machine learning [22], and deep learning models [23,24]. The importance of these models has increased with the growth of data and the availability of computing power, helping medical diagnostics and clinical decision making [25,26]. The objective of MoCab is to streamline the utilization of patient data for clinical decision making, with models stored within an easily accessible "cabinet" and readily retrievable as needed. Furthermore, MoCab adheres to the FHIR standard, ensuring that it integrates seamlessly with various EHR systems [27,28]. The architecture features a modular design for adaptability and scalability, supports continuous model training and updating, and includes a data management module that autonomously acquires and processes new data from EHR systems, thus maintaining the accuracy and relevance of the models. Ultimately, MoCab is designed to integrate with clinical workflows, improving the effectiveness of predictive models to support clinical decision making and improving patient care outcomes.

The paper is structured as follows. Section 2 addresses the challenges of applying predictive models in EHR systems and related work. Section 3 describes the architecture of the proposed MoCab, highlighting its modular design, integration capabilities, and continuous model training features. Section 4 describes the integration of MoCab with clinical decision support systems. Section 5 illustrates the practical implementation and demonstration of MoCab's use of predictive models. The paper concludes with Section 6, which discusses the implications of the findings, the limitations of the current framework, and future directions to improve MoCab's capabilities.

## 2. Related work

In recent years, tools and platforms have been developed to harness the potential of predictive modeling in healthcare, and the advantages and disadvantages are summarized in Table 1.

### 2.1. Web-based platforms for predictive modeling

EPOCH® and ePRISM® are examples of such web-based platforms designed to assist healthcare professionals in the creation and use of predictive models [29]. Initially, these platforms were designed to bridge the gap between complex prognostic models and clinical application, facilitating real-time decision support directly at the point of care. ePRISM® faces challenges in EHR integration due to its dependence on manual data entry, a time consuming and error-prone process that complicates integration with clinical workflows.

### 2.2. FHIR as a facilitator of interoperability

The FHIR standard, developed by Health Level Seven International (HL7), has been pivotal in promoting data exchange across HISs, thus enhancing interoperability. FHIR employs a RESTful approach to data integration, which is simple, agile, and adaptable, facilitating a smoother and more efficient healthcare information exchange process [32]. The bottom-up and consumer-based development model, different from the previous HL7 v2 and v3 versions, is another advantage of the FHIR standard [32]. This standard has been instrumental in supporting mobile health applications, which use FHIR to provide patients with access to their medical records, thus promoting patient participation and self-management [33]. Furthermore, FHIR has also been instrumental in the development of new applications and services in healthcare [34]. For example, platforms such as KETOS use FHIR to enable researchers to perform statistical analyses and develop, train, and deploy machine learning models in various EHR systems [35]. In addition, Khalilia et al. [30] have demonstrated the use of FHIR web services for the development and deployment of clinical predictive modeling. Their research outlines a framework for integrating predictive analytics into clinical workflows using FHIR standards. A systematic review concluded that the FHIR standard can provide an optimized solution to address health data interoperability issues, with the support of smart technologies, which solve the limitations of previous standards [36]. Furthermore, previous research has shown how FHIR could facilitate the transformation of healthcare data over 5G network slices. This capability is important for real-time health monitoring and is expected to be significant in future healthcare delivery models [37]. These studies confirm that the robust and flexible architecture of FHIR significantly contributes to advancements in healthcare interoperability, data management, and the acceleration of digital health innovations.

### 2.3. Challenges of model integration in HISs

Despite these advances, integrating multiple predictive models into HISs introduces challenges such as "app fatigue" among physicians due to the abundance of available models and applications [38]. With a growing number of models and applications, healthcare providers face the difficult task of discerning the optimal timing and context of their use. This challenge is further exacerbated by the abundant functionalities offered by platforms such as KETOS. CDS Hooks and SMART on FHIR address these issues. CDS Hooks provides advanced alerting and clinical decision support by analyzing patient data for specific risk factors, ensuring timely recommendations [39]. SMART on FHIR offers an interoperable app platform for EHRs, facilitating standardized patient data access across different FHIR-based EHR systems [40].

### 2.4. Handling large data volumes with FHIR

The standard FHIR API, while effective, struggles with the voluminous data demands of model training. A previous research [30] highlights that FHIR server response times increase dramatically as the volume of patient data requests grows, posing a significant challenge for efficient data retrieval. This performance bottleneck can hinder

**Table 1**
Comparison of the related works.

| Authors | System or Framework | Advantages | Disadvantages |
|---|---|---|---|
| Soto et al. [29] | EPOCH® and ePRISM® System | • Web-based decision support system. | • Manual data entry.<br>• Lack of integration of the API with existing EHR data. |
| Khalilia et al. [30] | Framework | • Support multiple model training algorithms.<br>• Use standardized data.<br>• Provide a REST interface. | • Focus on pre-determined algorithms.<br>• No support for custom algorithm development.<br>• Lack of automatic model retraining capabilities. |
| Gruendner et al. [31] | KETOS System | • Use standardized data.<br>• Allow researchers to train and deploy statistical models in a clinical research environment.<br>• Protects patient privacy while accessing large data sets. | • Lack of support for CDS Hooks and SMART on FHIR, limiting seamless integration with some EHR systems.<br>• Lack of automatic model retraining capabilities. |

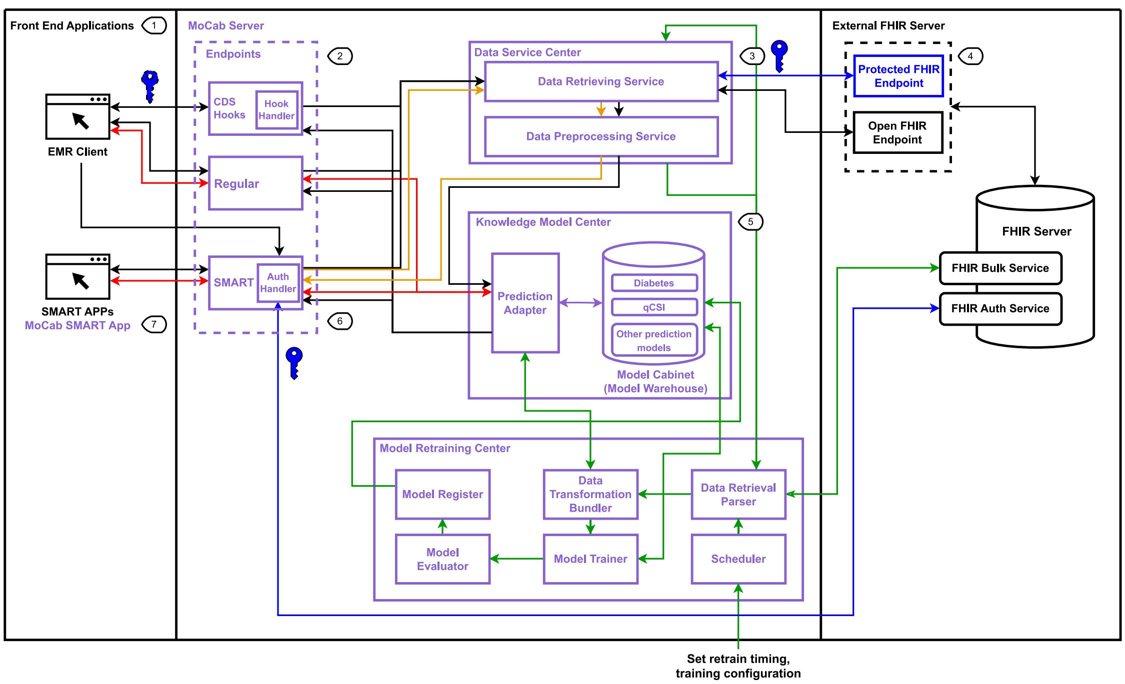


**Fig. 1.** MoCab architecture. The black lines represent the process of applying the model using patient identification to retrieve data from FHIR server, while the red lines indicate the process with data input by users or other applications. The green lines represent the continuous training mechanism. The yellow lines highlight the retrieval process of historical patient data for the SMART on FHIR App. The blue lines delineate the SMART authentication process, which is critical for secure data access. The blue keys represent the SMART OAuth token, which is utilized to protect access to the FHIR server. The numbers are sequences of the steps of the model application process.

the timely development and deployment of machine learning models within EHRs. The bulk FHIR service mitigates this by allowing efficient retrieval of large datasets from FHIR servers, crucial to the development and deployment of machine learning models within EHRs [16].

## 3. System architecture

The overall architecture of MoCab, described in Fig. 1, performs primarily three functions. First, the **Data Service Center** retrieves patient data from the FHIR server, ensuring that the appropriate information is accessed and prepared for analysis. Second, these data are then transferred to the predictive model in the **Knowledge Model Center**. Moreover, the architecture includes a **Model Retraining Center**, ensuring that the models are regularly updated and refined over time, thus maintaining their accuracy in ever-evolving clinical settings. In addition to these core functionalities, MoCab integrates both CDS Hooks and SMART on FHIR endpoints. This integration is to improve clinical decision-making processes. CDS Hooks serve a dual purpose, proactively alerting healthcare professionals to potential risks and allowing real-time tracking of health indicators. The source codes of MoCab are available at GitHub (https://github.com/DHLab-TSENG/MoCab).

### 3.1. Software architecture

The MoCab uses the 4+1 view model [9], including logical, development, process, physical views, and scenarios, to fully illustrate the structure and functionality of the system.

The logical view focuses on the main functional components of the system, including the user interfaces (such as the EMR client and the SMART app), the model developer, and others. The user interfaces interact with MoCab endpoints, which communicate with various centers and services for data processing, model management, and authentication. The configuration service, managed by the model developer, configures the data service center, the knowledge model center, and the model retraining center, with the latter two retrieving data from the FHIR server (Fig. 2).

The process view addresses the dynamic aspects of the system, highlighting key processes such as data retrieval and prediction. Healthcare professionals request predictions, initiating a multi-step process from data fetching to result delivery. Model retraining, triggered by a scheduler, involves data retrieval, preparation, training, evaluation, and registration.

The physical view describes the system's deployment, including the FHIR server, MoCab server, EMR client, and the network connections

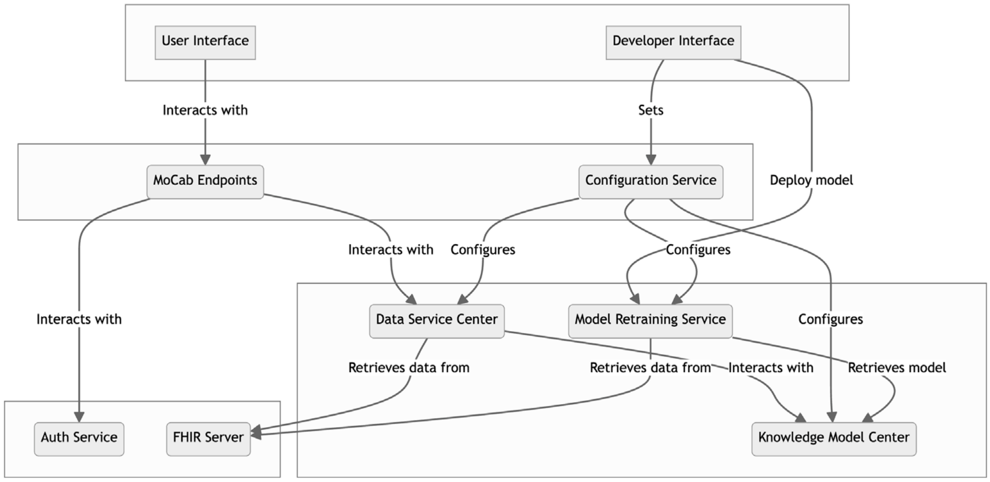


**Fig. 2.** The logical view of MoCab architecture.

**Table 2**
Example of the feature table.

| Model | Feature | Code | Code system | Type of data | Data alive time | Default value | Search type | Value route | Date route |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Body-temp | 8310-5 | http://loinc.org | Observation | 0007-00-00T00:00:00 | 36 | latest | – | – |
| 1 | Body-temp | 8331-1 | http://loinc.org | Observation | 0007-00-00T00:00:00 | 36 | latest | – | – |
| 2 | Hypertension | I10 | https://icd.who.int/browse10/2019/en | Condition | 0007-00-00T00:00:00 | – | latest | – | – |
| 2 | Margin of surgery | 90 | https://mitw.dicom.org.tw/IG/TWCR_LF | Procedure | 0007-00-00T00:00:00 | – | latest | Margin of surgery | – |

between the servers and the clients to build this system. It also includes middleware components like authentication service.

Lastly, the scenarios illustrate specific use cases, differentiating between user interactions and the self-training processes of MoCab. For users, scenarios such as accessing medical records and opening the SMART App activate the MoCab process. For model training, the process is triggered by the time interval defined by the model developer.

**Table 3**
Example of the resource route table.

| Conditions | Methods |
|---|---|
| margin_of_surgery | Procedure.bodySite.0.coding.0.code |
| gender | Patient.gender |
| age | Patient.get_age() |
| encounter_type | Encounter.class.code |

### 3.2. Data service center

The Data Service Center comprises two main services: the **Data Retrieval Service** (DRS) and the **Data Extraction Service** (DES) (Fig. 1).

#### 3.2.1. Data Retrieval Service (DRS)

The DRS is responsible for fetching data from the FHIR server with the REST API (Fig. 3). Each feature required by the predictive model is listed in a "feature table" (Table 2), including specific codes and the eligible time frame for the records. For example, hypertension configuration uses code `I10` in the ICD-10 code system, and to extract data covering the last seven years, using a designated `data_alive_time` setting "0007-00-00T00:00:00". The DRS has the ability to handle multiple coding scenarios for a single feature. Furthermore, when multiple records are available, the DRS uses specified search criteria (e.g., latest, maximum, or minimum), defined in `search_type`, to select the most relevant record. For example, in Table 2, LOINC 8310-5 and 8331-1 both refer to body temperature, and the `latest` prefix enables DRS to obtain the latest data. In cases where the data do not match the query, the service returns a default value in `default_value` or an API error, depending on the configuration in the feature table.

#### 3.2.2. Data Extraction Service (DES)

After retrieval, the DES takes control and extracts data from the retrieved resources to fit the format conducive to model input (Fig. 4). DES extracts specific values and associated timestamps from these resources, tailoring data extraction to the nature of the resource. For example, observation resources contain measurement results and observation dates, whereas condition and procedure resources are processed into Boolean values and corresponding dates, representing whether the server returns resources, and the date is the time of the latest recorded data. In addition, patient resources can be utilized to calculate the patient's age.

Features can also be extracted by the path of resources defined by users. For example, the margin of surgery is stored in the "bodysite" of the procedure resources based on the definition of the Taiwan Cancer Registry team [41]. The resource route table (Table 3) indicates the extraction path for data stored in any column within a resource. This ensures data extraction, even for complex data structures. Finally, after extracting all the necessary features, the extracted data can be packed and sent to the Knowledge Model Center.

### 3.3. Knowledge model center

Upon retrieval and extraction of the data by the Data Service Center, MoCab directs these data to the Knowledge Model Center, which consists of **Prediction Adapter** and **Model Cabinet**.

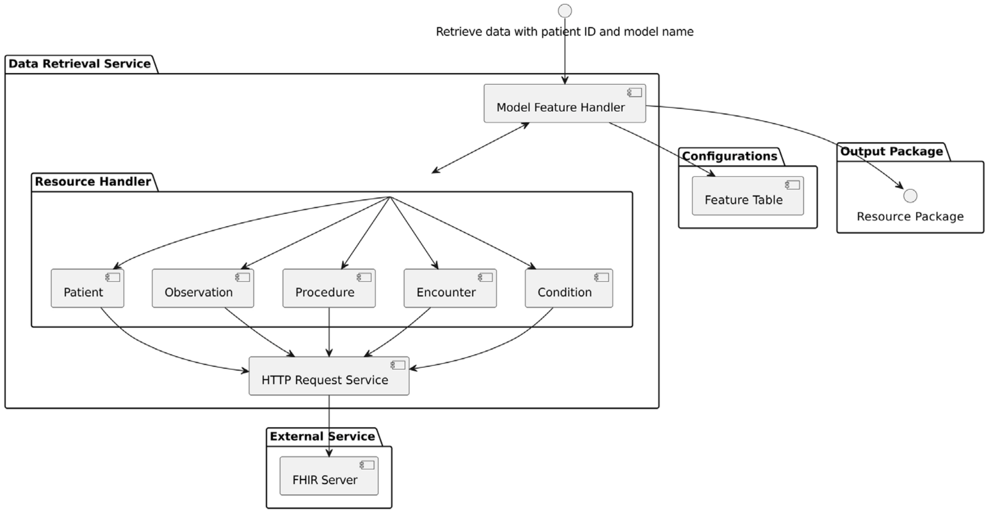


**Fig. 3.** Data retrieval service architecture.

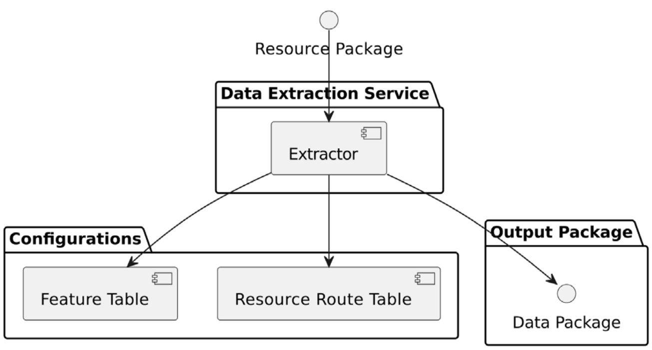


**Fig. 4.** Data extraction service architecture.

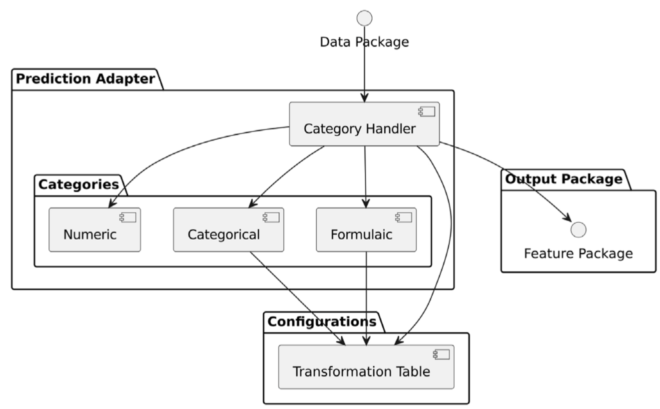


**Fig. 5.** Prediction adapter architecture.

**Table 4**
Example of the transformation table.

<table>
<tr><th>Model</th><th>Feature</th><th>Type</th><th>Formulate</th><th>Index</th></tr>
<tr><td>CHARM</td><td>Hypothermia</td><td>Category</td><td>1=lt | 36</td><td>1</td></tr>
<tr><td>CHARM</td><td>Hypothermia</td><td>Category</td><td>0=ge | 36</td><td>1</td></tr>
<tr><td>SPC</td><td>BMI_label</td><td>Category</td><td>1=BMI lt | 18.5</td><td>1</td></tr>
<tr><td>SPC</td><td>BMI_label</td><td>Category</td><td>2=BMI ge | 18.5& lt|24</td><td>1</td></tr>
<tr><td>SPC</td><td>BMI_label</td><td>Category</td><td>3=BMI ge | 24</td><td>1</td></tr>
<tr><td>SPC</td><td>BMI</td><td>Formula</td><td>[weight]/([height]/100)<br>/([height]/100)</td><td>1</td></tr>
</table>

### 3.3.1. Prediction adapter

Before importing data into models, we need to adapt the data to formats compatible with different models. For example, some models accept numeric inputs, such as white blood cell count, while others require nominal data, such as an indicator of overweight or underweight, or hypothermia in patients. In addition, certain features, such as BMI, require further calculations based on the original records. To efficiently address these varying input format requirements, MoCab incorporates a Prediction Adapter module, which converts features into the model-specific preferred input format (Fig. 5). A “transformation table” (Table 4) is used to outline the data transformation configurations, enhancing the adaptability of the system to various model requirements.

Within MoCab, features are categorized into three types: numeric, categorical, and formulaic. Numeric features are directly fed into the model without transformation. In contrast, categorical and formulaic types are transformed based on user-defined rules. The formulaic type facilitates operations on one or more features to derive a new feature. Subsequently, the transformed data are compiled into an array ready for model prediction, with each feature indexed according to the transformation table (Table 4).

The Prediction Adapter extends its capabilities to handle pre-processed records (Fig. 1, red lines). It verifies the completeness of these records before they are transmitted to the predictive model, thus ensuring the integrity and applicability of the data used in models stored in the Model Cabinet.

### 3.3.2. Model cabinet

Model Cabinet is a dedicated repository for storing the models along with their respective dependencies. Each model is stored in its folder that contains a ‘predict’ function (Backend/mocab_models/template/). This function is integral to the prediction process, where it receives input from the Prediction Adapter, makes the prediction, and relays the prediction result back to MoCab. The results, along with the processed data, are then sent to the end-user through an API.

To facilitate model integration, MoCab provides a template compatible with models exported from frameworks such as Joblib or TensorFlow (Backend/mocab_models/template/). This template enables seamless model replacement and updating; it requires only substituting the trained model and updating the model’s name to ‘register_model’. Additionally, the necessary data configurations, including features and transformation tables, are integrated, making the model ready for prediction.

**Table 5**
Example of the training parameter table.

| Model | Filter | Interval | Null value strategy | Test size | Validation index |
|---|---|---|---|---|---|
| 1 | (date)lt \| [hypertension] | 0001-00-00T00:00:00 | (drop)gt \| 3&(mean) | 0.2 | Accuracy |
| 2 | (date)lt \| [target_1] | 0000-06-00T00:00:00 | (drop)gt \| 5&(median) \| [feature_1]&(mode) | 0.2 | AUROC |

### 3.4. Model retraining center

The concept of continuous training is crucial in machine learning, particularly to maintain the accuracy of models in dynamic environments [12,13]. This process, designed based on MLOps principles, involves periodically updating the model using transfer learning with new data to account for possible changes in the underlying data patterns or operational context [6]. The Model Retraining Center synthesizes the retraining process into six modules [42]: **Scheduler**, **Data Retrieval Parser**, **Data Transformation Bundler**, **Model Trainer**, **Model Evaluator**, and **Model Register**. This approach ensures that the models integrated within MoCab remain up-to-date and accurate, providing reliable predictions in a clinical setting.

#### 3.4.1. Scheduler

The fundamental mechanism for configuring this Scheduler is the training parameter table (Table 5). By specifying the time interval for retraining, the Scheduler is programmed to initiate the continuous training pipeline at predetermined time intervals.

#### 3.4.2. Data retrieval parser

The Data Retrieval Parser extracts data from the FHIR server and parses them into a format suitable for model training. To facilitate the retrieval of large datasets, this parser uses the FHIR bulk service [16]. Upon issuing a 'kick-off' request, the FHIR bulk service returns data in Newline Delimited JSON (NDJSON) format, which complies with the FHIR specification. The subsequent stage involves data retrieval and extraction from these resources. The configuration of required features is established in the Data Service Center (Fig. 1, green line linked to the Data Service Center). This feature table allows MoCab to identify relevant features or prediction targets by matching them with corresponding codes and code systems via DRS (Table 2). The DES then extracts specific values from these resources. The temporal alignment of the data is also ensured by further configurations. For example, in a model designed to predict hypertension, it is imperative to use the data recorded before the appearance of hypertension. MoCab addresses this by setting appropriate time intervals between features and prediction targets in the 'Filter' field of the training parameter table (Table 5, model 1). This approach to temporal data alignment is vital to the rationality of the training process.

#### 3.4.3. Data transformation bundler

The Data Transformation Bundler transforms input features and targets into appropriate formats and bundles these data, then splits them into training and testing sets. Similar with the transformation steps in the Knowledge Model Center, the Data Transformation Bundler utilizes the Prediction Adapter to transform the data (Fig. 1, green line linked to the Data Transformation Bundler). This process is governed by the specifications described in both the transformation table (Table 4) and a corresponding table for the targets, which allows the integration of patient data into a cohesive dataset.

The next step is to exclude patients who are not the study target. For example, as demonstrated in the second primary cancer prediction model for lung cancer survivors [43], patients who did not experience lung cancer as their first primary cancer were deemed non-target for the model and subsequently excluded. For records with missing values, developers can specify the threshold for exclusion or select appropriate imputation methods (e.g., mode, mean, median, or specific values) in the training parameter table (Table 5, null_value_strategy). This flexibility allows for customized strategies for different features (Table 5, null_value_strategy of model 2).

MoCab then randomly divides the data set into training and testing subsets. This division is based on the test size configuration detailed in the training parameter table (Table 5, test_size). The training data is then funneled into the Model Trainer, while the testing data is reserved for the Model Evaluator. An additional preparatory step, particularly for models dealing with categorical variables, involves encoding these variables into a binary format (dummy or indicator variables). This encoding process is executed before the distribution of datasets to the subsequent modules.

#### 3.4.4. Model trainer

The Model Trainer module fine-tunes models with new data by retrieving the original model and retraining configurations (Backend/mocab_models/) and retraining with the latest training data. The model architecture will follow the settings of the original model. For example, a deep neural network architecture is used to predict second primary cancer (SPC). Then, the retraining process involves defining optimization algorithms with the "train" function and configuring essential hyperparameters in the configuration file, such as learning rates, batch size, epochs, and evaluation thresholds. The system then initiates the training function, which fine-tunes the model using a transfer learning approach and stores the retrained model in its respective directory. To streamline the training process, MoCab offers a specialized continuous training template (Backend/mocab_models/SPC/), simplifying the training procedure by allowing users to replace the training compiler. Once training is completed, the updated model is saved under the designation "new_model" for further evaluation in the Model Evaluator module.

#### 3.4.5. Model evaluator

Recognizing that a fine-tuned model may not necessarily outperform the original one, MoCab employs an evaluation process using the test data provided by the Data Transformation Bundler. This evaluation process focuses on the metrics computation to determine the performance of existing (original) and newly fine-tuned (new) models. Key metrics such as accuracy, F1 score, and the area under the receiver operating characteristic curve (AUROC) are used for comparison (Table 5, validation_index). These metrics were selected for their broad acceptance in the field and ability to assess model performance. They capture overall accuracy, the balance between precision and recall, and the model's discriminatory power, respectively, making them suitable for model evaluation. The selection of the most effective model is based on the validation_index specified in the training parameter table (Table 5).

#### 3.4.6. Model register

This final step involves integrating the model that demonstrated higher performance during the evaluation phase into the Model Cabinet (Fig. 1, green line linked to the Model Register). The Model Register also documents the metadata for the model, the number of patients involved in the training set, the timestamps of the last training and data acquisition, and the evaluation results (Table 6). These records provide information on the training history, facilitating transparency and traceability of model development.

**Table 6**
Example of the training status table.

| Model | Last training time | Last training data time | Num of pat | Old model evaluate | New model evaluate | Register model | Threshold |
|---|---|---|---|---|---|---|---|
| 1 | 2023-05-10 | 2018-01-01 | 481 | 0.683 | 0.665 | Register | 0.12 |
| 1 | 2023-06-10 | 2018-11-08 | 481 | 0.696 | 0.741 | New | 0.061 |

**Table 7**
Example of the CDS hooks table.

| Model | Condition | Info range | Warning range | Critical range |
|---|---|---|---|---|
| qCSI | encounter_type=EMER | le\|4 | gt\|4&le\|9 | ge\|9 |
| NSTI | encounter_type=EMER | lt\|0.3 | ge\|0.3&lt\|0.7 | gt\|0.7 |

## 4. Integration with clinical decision support system

MoCab offers three interaction modalities, which are open FHIR endpoints, MoCab regular endpoints, and CDS Hooks endpoints. For entities that utilize open FHIR endpoints or their methods for data retrieval, the MoCab regular endpoint (Fig. 1, Regular Endpoint) is tailored to their needs. To combat app fatigue caused by the proliferation of CDS services, MoCab incorporates a strategic approach with CDS Hooks, which is designed to generate predictions for at-risk patients and guide the delivery of appropriate alerts to physicians (Fig. 1, CDS Hooks Endpoint) [38]. We also provide the option to activate MoCab from SMART on FHIR App (Fig. 1, SMART Endpoint).

Each model within MoCab caters to a specific use scenario. The CDS Hooks table specifies the contextual usage of each model (Table 7). For example, the CHARM model [44] is designed primarily to predict mortality among patients with suspected sepsis for emergency department applications, and an emergency department encounter in FHIR is coded as "EMER" [45]. Accordingly, the CDS Hooks table can define a triggering condition as "encounter_type=EMER", indicating that MoCab will activate the model for records originating from emergency departments. The sources of conditions are in the resource route table (Table 3). The "encounter_type" is mapped to "class.code" in the FHIR Encounter resources. MoCab extracts this information to determine the applicability of a model to a specific patient scenario. Additionally, for conditions requiring computations, such as patient's age, the necessary function is specified in the resource route table (Table 3, "age" condition).

A notable feature of CDS Hooks within MoCab is their ability to deliver customized alerts to physicians. CDS hooks generate CDS cards, which encompass a range of information, from basic text notifications (information card) and alternative action suggestions (suggestion card) to links that lead to applications or reference materials (app link card). The cards are categorized into three indicators – info, warning, and critical – each signifying the level of urgency or importance of the information presented.

In MoCab, the CDS Hooks endpoint is configured to return different cards based on the risk thresholds defined for each model (Table 7). These cards not only display the model's risk score and descriptive results but also include a SMART app link. Activating this link (Fig. 1, black line from EMR client to the SMART Endpoint) initiates a request for SMART on FHIR authentication from the EHR server to secure access permission, allowing the MoCab SMART app to retrieve the data of the targeted patient from the FHIR server.

The MoCab SMART app integrates the core functions of MoCab, such as risk calculation, with additional features. A significant feature is visualizing the historical patient data through trend charts, accessible through the SMART endpoint (Fig. 1, orange line). These charts provide physicians with a view of patient health over time, improving the depth and quality of clinical decision-making.

## 5. Implementation and demonstration

To demonstrate the practicality of MoCab, we implemented the framework using three different types of models: a scoring model (the quick COVID severity index [46]), a machine learning model (identifying necrotizing soft tissue infections (NSTI) [47]), and a deep learning model (predicting SPC in patients with lung cancer [43]). To simulate realistic data scenarios, we generated synthetic data that are consistent with the structure of one of the largest EHR database, the Chang Gung Research Database (CGRD) [48], supplemented by examples from the official FHIR documentation [49]. This section delineates the implementation steps, including the generation of FHIR resources, the deployment of models, and interactions with MoCab to obtain prediction results. The source codes of the models are available on GitHub [50].

### 5.1. Data and FHIR server

Recognizing the growing prominence of FHIR in the landscape of EHR [36,51], we adopted FHIR version R4 as the standard protocol for data retrieval to create a system compatible with a variety of EHRs. We use HAPI FHIR [52], version 6.4.0, for its robust support of FHIR bulk services.

Data used in this study was simulated based on CGRD [48], then converted into FHIR format with the FHIR Extract Transformation Load (ETL) tool [53]. This process was guided by the official FHIR documentation [54] and was supplemented with information from the FHIR Implementation Guide (IG) documents [55]. Fig. 6 illustrates the key attributes of the data in FHIR format and Table 8 describes it in detail. The Chang Gung Medical Foundation Institutional Review Board approved this study (IRB no. 201901386B0) and waived the requirement of patient consent.

### 5.2. Quick COVID Severity Index (qCSI)

The Quick COVID Severity Index (qCSI) is for rapidly assessing COVID-19 patients in emergency settings [46]. The model translates the respiratory rate, $SpO_2$ (oxygen saturation), and $O_2$ flow rate using a scoring system, where the cumulative score of these characteristics estimates the risk of critical respiratory illness. This scoring mechanism helps physicians make informed clinical decisions. For example, a high qCSI score could indicate the need for immediate respiratory support interventions, such as high flow nasal cannula or mechanical ventilation, thus optimizing resource allocation and potentially improving patient outcomes.

For implementation, we refer to the format of the vital sign profile on the official FHIR website [54]. Subsequently, we extracted vital signs from the CGRD, converting this information into FHIR resources. It is important to note that the $O_2$ flow rate data, originally recorded in plain text, were assigned to the valueString field in FHIR. Details of this data mapping are given in Table 9.

The feature retrieval process is based on the feature table (Backend/config/features.csv), which specifies the CGMH code systems for respiratory rate and $SpO_2$, along with LOINC terminology codes for $O_2$ flow rate. All data used were recorded within one hour. Subsequently, we defined the data transformation configurations (Backend/config/transformation.csv) based on the variables of the qCSI scoring model [46]. In this phase, the Prediction Adapter converts numeric values into corresponding qCSI points according to the scoring model's criteria. The culmination of this process involves the summation of

**Table 8**
Description of the example of FHIR observation data.

| Key | Value | Description |
|---|---|---|
| resourceType | Observation | The type of resource, in this case, an Observation |
| id | oxygen-saturation3 | Unique identifier for the observation |
| coding | – | List of codes defined by terminology systems |
| system | https://www.cgmh.org.tw<br>http://loinc.org<br>http://loinc.org | Terminology system URL for the coding info |
| code | OA08<br>2708-6<br>59408-5 | Code for the coding |
| display | Pulse Oximetry<br>Oxygen saturation in Arterial blood<br>Oxygen saturation in Arterial blood by Pulse oximetry | Display text for the coding |
| subject_reference | Patient/test-03121002 | Reference to the patient associated with the observation |
| effectiveDateTime | 2019-07-02 | Clinically relevant time for observation |
| valueQuantity | "value": 97<br>"unit": "%O2"<br>"system": "http://unitsofmeasure.org"<br>"code": "%" | Actual result of the observation, including its value, unit, terminology system of unit, and code of the unit |

**Table 9**
Data mapping table in qCSI.

| Data model of CGRD | FHIR resource name | FHIR resource attribute | Description |
|---|---|---|---|
| Patient_id | Patient | id | The identifier for the patient in the hospital |
| Patient_id + "_" + inpatient_id | Encounter | id | The identifier for the encounter of the patient in the hospital |
| Body Site | Observation | bodySite.code | The body site of the observed part |
| Code | Observation | code.coding.code | The vital sign code in the CGRD code system or LOINC code |
| Respiratory rate | Observation | valueQuantity.value | Respiratory rate |
| $SpO_2$ | Observation | valueQuantity.value | $SpO_2$ |
| $O_2$ flow rate | Observation | valueString | The $O_2$ flow rate of the ventilator |

**Table 10**
Data mapping table in the NSTI model.

| Data model of CGRD | FHIR resource name | FHIR resource attribute | Description |
|---|---|---|---|
| Patient_id | Patient | id | The identifier for the patient in the hospital |
| Birthdate | Patient | birthDate | The birthdate of the patient |
| Patient_id + "_" + inpatient_id | Encounter | id | The identifier for the encounter of the patient in the hospital |
| Body Site | Observation | bodySite.code | The body site of the observed part |
| Code | Observation | code.coding.code | The laboratory feature codes in the CGRD code system |
| White blood cell | Observation | valueQuantity.value | White blood cell |
| C-Reactive protein | Observation | valueQuantity.value | C-Reactive protein |
| Segment | Observation | valueQuantity.value | Segment |
| Band | Observation | valueQuantity.value | Band |

points from each feature, upon which MoCab returns the final qCSI score. Another challenge was handling the $O_2$ flow rate data stored as plain text on the FHIR server. To address this, we implemented regular expression parsing to extract specific values from the text.

### *5.3. Necrotizing Soft Tissue Infection risk model (NSTI)*

Necrotizing fasciitis, a rare but critical infection that affects soft tissues, often requires immediate treatment to prevent fatal outcomes [56]. NSTI prediction is critical for timely and appropriate treatment to prevent fatal outcomes. We included an NSTI risk model, built with logistic regression [47], to differentiate necrotizing fasciitis from other soft tissue infections. This model is particularly valuable in the emergency department in coastal regions with prevalent waterborne bacterial infections. The NSTI model requires input on five features for analysis: seawater exposure and four laboratory results: white blood cell count, C-reactive protein, creatinine, and sodium (Table 10). However, the CGRD does not include records of seawater exposure. To address this, physicians can manually indicate seawater exposure in the MoCab SMART App. By integrating these inputs, the NSTM model facilitates early diagnostic decisions, potentially leading to faster surgical interventions, which are often lifesaving.

The feature table presents the data retrieval configurations for the NSTI model (Backend/config/features.csv). The transformation table delineates the complete data transformation configuration (Backend/config/transformation.csv). Subsequent to the data transformation stage, these features are sent to the final model imported in Joblib format in Knowledge Model Center - Model Cabinet.

### *5.4. Second Primary Cancer risk model (SPC)*

SPC occurs when an individual develops a new primary cancer following an earlier primary occurrence [57]. Factors such as specific cancer treatments, genetic predispositions, and exposure to cancer-causing substances significantly increase the risk of SPC [58,59]. Lung cancer, which is notably prevalent and widespread worldwide, often requires careful monitoring of SPC occurrences [60,61]. This SPC risk prediction model is especially relevant in clinical follow-up programs,

```
{
  "resourceType": "Observation",
  "id": "oxygen-saturation3",
  "code": {
    "coding": [
      {
        "system": "https://www.cgmh.org.tw",
        "code": "OA08",
        "display": "Pulse Oximetry"
      },
      {
        "system": "http://loinc.org",
        "code": "2708-6",
        "display": "Oxygen saturation in Arterial blood"
      },
      {
        "system": "http://loinc.org",
        "code": "59408-5",
        "display": "Oxygen saturation in Arterial blood by
                    Pulse oximetry"
      }
    ]
  },
  "subject": {
    "reference": "Patient/test-03121002"
  },
  "effectiveDateTime": "2019-07-02",
  "valueQuantity": {
    "value": 97,
    "unit": "%O2",
    "system": "http://unitsofmeasure.org",
    "code": "%"
  }
}
```

**Fig. 6.** An Example of FHIR Observation Data. The code attribute indicates the type of observation, including “OA08” for “Pulse Oximetry” in the CGRD terminology system and “2708-6” LOINC code for oxygen saturation. The subject references the patient with the ID “Patient/test-03121002”. The effectiveDateTime specifies the observation date as 2 July 2019. ValueQuantity records the measured oxygen saturation as 97.0 with unit % $O_2$. These attributes ensure a precise and standardized representation of the data. Other examples are in https://github.com/DHLab-TSENG/MoCab/tree/main/Configs/output/Test Data.

where SPC risk monitoring can guide preventive measures. This model can be used in the surveillance of lung cancer survivors who have undergone radiation therapy, a group at elevated risk of developing cancers in the radiation field. By predicting the risk of SPC, the model helps tailor patient-specific surveillance protocols, improving personalized care and improving outcomes. We trained an SPC prediction model for lung cancer survivors using TensorFlow, utilizing the data generated based on the cancer registry dataset in CGRD [43]. The model, which requires 27 features, integrates multiple FHIR resources, including patient, encounter, condition, observation, and procedure.

To create data in FHIR format, we followed the guidelines of the Taiwan Cancer Registry Center (TWCR) [41], the minimal Common Oncology Data Elements (mCODE) [62], and insights from the Taiwan ‘cancer2fhir’ team [63]. This approach facilitated conversion of the cancer registry dataset. Data retrieval and transformation configuration details are provided on GitHub (Backend/config/features.csv and transformation.csv).

For continuous training, we set specific parameters: (1) Features needed to be time-stamped between the first and second primary cancers (Table 11, Filter). (2) Records with more than three missing values were excluded. (3) Missing values were imputed with the median (Table 11, Null value strategy). (4) 20% of the data was selected as testing set. (5) The AUROC was used for validation (Table 11, Test size and Validation index). The feature table and the transformation table for the target configuration are available on GitHub (Backend/config/continuous_training/), and users can adjust these settings by modifying the configuration tables. For example, the last training data was originally 2018-01-01 (Table 12, first row). After continuous training, the training status table was updated to reflect the latest training results (Table 12, second row).

### 5.5. Integrations with CDS hooks and SMART on FHIR

To simulate the interaction between HIS and the MoCab framework, experiments were carried out using CDS Hooks and SMART on FHIR Sandbox [64]. The configuration of the CDS Hooks is in Table 7. This setup includes predefined warning ranges for each model, enabling the system to issue appropriate alerts. For example, in the case of the qCSI model, the system is configured to present an information card for scores less than 4, a warning card for scores between 4 and 9, and a critical alert for scores exceeding 9 points (Table 7, qCSI). For the NSTI and qCSI models, designed primarily for emergency department use, the models are automatically activated upon the onset of an emergency department encounter. In preparation, we identified the required condition routes, as specified in the CDS Hooks scenarios. These routes were documented in the resource route table (Backend/config/resource.route). Taking the qCSI model as an example, the patient’s encounter resource was retrieved following the model requirements and the definition of the encounter_type condition in the resource route table. Upon verification that the type of patient encounter matched the criteria, MoCab initiated the qCSI model prediction process. This simulation demonstrated MoCab’s capability to trigger accurately and process model predictions based on specific criteria, thereby validating its practical utility.

One of the key contributions of MoCab is the creation of an intuitive SMART on FHIR application, designed to facilitate the interaction between physicians and the model in MoCab. The application is accessible through the ‘MoCab-App’ button on the CDS Hook card. Once activated, it presents a user interface that displays key parameters involved in the predictive analysis of the model (Fig. 7). In addition, this interface offers a comprehensive overview of the patient and trend analysis, significantly helping physicians in their decision-making process. In particular, it allows for real-time adjustments to clinical features, enabling physicians to view updated prediction results, and empowering physicians by providing greater control and flexibility. Furthermore, the application is enhanced with additional functionalities like timeline filtering. These features are user-centric and provide valuable information, increasing the overall utility of the MoCab system in clinical settings.

## 6. Discussion and conclusion

MoCab offers numerous advantages, particularly in reducing the costs associated with model deployment in various EHR systems with the FHIR standard, supporting continuous training to maintain model performance over time, and improving decision-making processes with CDS Hooks and SMART on FHIR.

This framework is primarily designed for healthcare professionals, researchers, and HIS administrators. Healthcare professionals can seamlessly integrate MoCab into their clinical workflows to improve patient care quality and precision in diagnostic and therapeutic decisions. As we mentioned in the session 5, the use of the qCSI model within MoCab during the COVID-19 pandemic can help emergency care providers quickly identify patients at high risk of critical respiratory illness and launch immediate interventions. In emergency and surgical care, the NSTI risk model in MoCab facilitates timely and potentially life-saving surgical interventions. Finally, the SPC prediction model in MoCab helps oncology professionals develop personalized surveillance plans, thereby improving patient care and long-term outcomes. Researchers benefit from MoCab’s capability to quickly incorporate new predictive

**Table 11**
Training parameters for SPC model.

| Model | Filter | Interval | Null value strategy | Test size | Validation index |
|---|---|---|---|---|---|
| SPC | (date)lt\|[seq_2]&(date)ge\|[seq_1] | 0001-00-00T 00:00:00 | (drop)gt\|3&(median) | 0.2 | AUROC |

**Table 12**
Training status table of SPC model after continuous training process executes.

| Model | Last training time | Last training data time | Numbers of patient | Old model evaluate | New model evaluate | Register model | Threshold |
|---|---|---|---|---|---|---|---|
| SPC | 2018-01-07 | 2018-01-01 | 235 | 0 | 0.696 | Register | 0.08 |
| SPC | 2018-11-10 | 2018-11-08 | 481 | 0.696 | 0.741 | New | 0.061 |

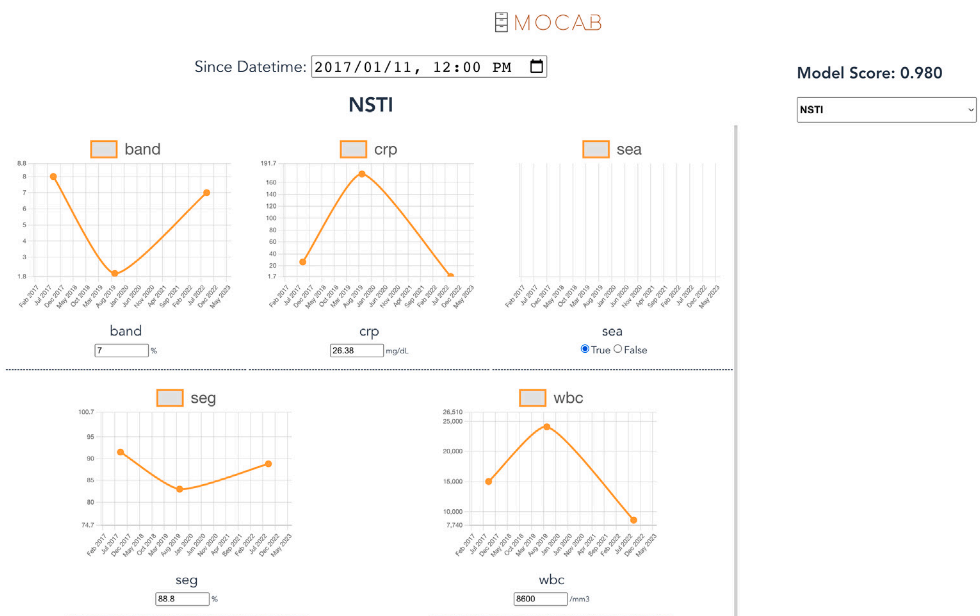


**Fig. 7.** MoCab SMART App user interface with the NSTI model. The "Since Datetime" label controls the time period of data we used for visualization and prediction. The orange line presents the trend of the patient's data, including laboratory results, vital signs, etc.

models, allowing for a quick transition from research findings to practical clinical applications. MoCab provides an infrastructure for HIS administrators to manage the predictive model lifecycle.

The MoCab system exhibits similarities with previous studies while introducing key enhancements. Similarly to the EPOCH® and ePRISM® systems [29], MoCab aims to apply machine learning models to assist clinical decision making. Furthermore, MoCab is consistent with the objectives of KETOS [31], which offers a solution integrated into HISs. However, unlike EPOCH® and ePRISM®, which require manual input of model parameters, MoCab capitalizes on the rapidly evolving FHIR standard. This utilization facilitates automatic data retrieval from EHR systems, thus obviating the need for manual parameter input. Furthermore, while KETOS lacks mechanisms for active alerts to healthcare professionals, MoCab incorporates CDS Hooks. This integration enables the delivery of varying levels of alerts to healthcare professionals. Furthermore, MoCab has the capacity for continuous learning. This feature allows the model to continuously train and refine its performance.

Despite MoCab's advantages, the framework encounters certain challenges and limitations. Primarily, MoCab has adopted the FHIR standard with the FHIR server. Although the FHIR standard facilitates standardized data interchange, it needs to be implemented in the HIS first. Currently, not all HISs are equipped to fully support the FHIR standard, which could slow the adoption of MoCab and limit its applicability. However, given the growing adoption of FHIR globally [36], we anticipate a reduction in this challenge, increasing MoCab's utility as FHIR becomes more prevalent in healthcare facilities. In addition, specific features may be unavailable on the FHIR server. To address this, MoCab is designed to accept pre-processed records and allows data to be gathered from users via the MoCab SMART App. Second, in the current framework, the performance comparison between updated and preexisting models is contingent on the performance metrics selected by the users. This approach, while practical, may introduce biases as it relies on a single data set for evaluation. Incorporating methodologies such as cross-validation and statistical tests is a possible solution to mitigate potential biases and improve the reliability of our model comparison. Third, environmental sustainability is another concern. A previous study suggests that Information and Communication Technology (ICT), including online services and electronic government, has the power to determine the world's ecological future in developed countries [65]. In addition, ICT could positively contribute to reducing $CO_2$ emissions once a level of ICT development has been achieved [66]. Although MoCab shows potential to advance predictive healthcare modeling, the current framework does not explicitly account for energy consumption and its environmental impact. The energy demands for model training are high [67]. Given the increasing emphasis on sustainable practices, integrating energy efficiency measures into MoCab's design is essential for future development [68]. Furthermore, a previous study has demonstrated innovative approaches in cloud computing environments that dynamically allocate resources to optimize energy efficiency without compromising system performance [69]. Future enhancements to MoCab will include energy efficiency metrics and dynamic resource allocation to mitigate environmental impact. Finally, to assess the performance of the MoCab framework, practical evaluations are needed in the future. Currently, the MoCab framework is designed primarily as a proof-of-concept to illustrate the potential and feasibility of automating a model management system within HISs. The primary aim at this stage is to validate the technical aspects, such as integration with EHR systems through FHIR standards, the operation of the Model Retraining Center, and the use of CDS Hooks in a controlled environment. These are critical to establish a foundational understanding of the framework for handling real-world healthcare data and workflows.

To address these challenges and limitations, immediate efforts will be made to expand its functionality by developing a user-friendly model import interface and providing customization options for encoding methods [70,71]. Furthermore, we will add more use cases to demonstrate MoCab's versatility and effectiveness. Mid-term goals will include integrating environmental sustainability into MoCab's main functions, allowing for real-time monitoring and management of computational resource usage. Longer-term initiatives will be to perform practical evaluations to determine the functionality and impact of MoCab in the real world and to refine the functionality and user interface of MoCab, ensuring that it meets the practical demands and regulatory requirements of real-world healthcare settings. These assessments will focus on user experience, system integration, and overall performance in clinical settings. Finally, the incorporation of the Clinical Quality Language (CQL) [72] is expected, further improving the integration capabilities of the model.

In summary, MoCab improves the interoperability of machine learning models across health information systems using the FHIR standard. Its Model Retraining Center dynamically updates models based on evolving clinical data. MoCab also incorporates CDS Hooks for proactive alert delivery within clinical workflows, improving decision-making without disrupting user experience. The modular design of the framework ensures adaptability, offering a promising solution to the challenges of adapting machine learning models in healthcare settings.

## Funding

This study was supported in part by grants from the National Science and Technology Council, Taiwan (NSTC 111-2628-E-A49-026-MY3), the Higher Education Sprout Project of the National Yang Ming Chiao Tung University and Ministry of Education (MOE), Taiwan (CGMH-NYCU-113-CORPG2P0071), and Chang Gung Memorial Hospital, Taiwan (CORPG2P0071). The funders had no role in the design and conduct of the study; collection, management, analysis, and interpretation of the data; preparation, review, or approval of the manuscript; and decision to submit the manuscript for publication.

## Code availability

The source codes for MoCab are available at GitHub https://github.com/DHLab-TSENG/MoCab/ .

## CRediT authorship contribution statement

**Zhe-Ming Kuo:** Writing – original draft, Software, Methodology, Data curation. **Kuan-Fu Chen:** Writing – review & editing, Funding acquisition. **Yi-Ju Tseng:** Writing – review & editing, Writing – original draft, Supervision, Resources, Project administration, Funding acquisition, Conceptualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

[1] E.H. Weissler, T. Naumann, T. andersson, R. Ranganath, O. Elemento, Y. Luo, D.F. Freitag, J. Benoit, M.C. Hughes, F. Khan, P. Slater, K. Shameer, M. Roe, E. Hutchison, S.H. Kollins, U. Broedl, Z. Meng, J.L. Wong, L. Curtis, E. Huang, M. Ghassemi, The role of machine learning in clinical research: transforming the future of evidence generation, Trials 22 (1) (2021) 537, http://dx.doi.org/10.1186/s13063-021-05489-x.

[2] X. Ren, X. Li, K. Ren, J. Song, Z. Xu, K. Deng, X. Wang, Deep learning-based weather prediction: A survey, Big Data Res. (ISSN: 2214-5796) 23 (2021) 100178, http://dx.doi.org/10.1016/j.bdr.2020.100178.

[3] F. Safara, A computational model to predict consumer behaviour during COVID-19 pandemic, Comput. Econ. 59 (4) (2022) 1525–1538.

[4] J.A. Sidey-Gibbons, C.J. Sidey-Gibbons, Machine learning in medicine: a practical introduction, BMC Med. Res. Methodol. 19 (2019) 1–18, http://dx.doi.org/10.1186/S12874-019-0681-4/TABLES/5.

[5] M.V. Maddali, M. Churpek, T. Pham, E. Rezoagli, H. Zhuo, W. Zhao, J. He, K.L. Delucchi, C. Wang, N. Wickersham, et al., Validation and utility of ARDS subphenotypes identified by machine-learning models using clinical data: an observational, multicohort, retrospective analysis, Lancet Respir. Med. 10 (2022) 367–377, http://dx.doi.org/10.1016/S2213-2600(21)00461-6.

[6] H. Akhlaghi, S. Freeman, C. Vari, B. McKenna, G. Braitberg, J. Karro, B. Tahayori, Machine learning in clinical practice: Evaluation of an artificial intelligence tool after implementation, Emerg. Med. Australas. n/a (n/a) (2023) http://dx.doi.org/10.1111/1742-6723.14325.

[7] K.D. Mandl, D. Gottlieb, J.C. Mandel, Integration of AI in healthcare requires an interoperable digital data ecosystem, Nat. Med. (2024) 1–4.

[8] J.P. Avendano, D.O. Gallagher, J.D. Hawes, J. Boyle, L. Glasser, J. Aryee, B.M. Katt, Interfacing with the electronic health record (EHR): A comparative review of modes of documentation, Cureus (ISSN: 2168-8184) 14 (6) (2022) e26330, http://dx.doi.org/10.7759/cureus.26330.

[9] S. Silvestri, A. Esposito, F. Gargiulo, M. Sicuranza, M. Ciampi, G. De Pietro, A big data architecture for the extraction and analysis of EHR data, in: 2019 IEEE World Congress on Services, SERVICES, Vol. 2642-939X, 2019, pp. 283–288, http://dx.doi.org/10.1109/SERVICES.2019.00082.

[10] A. Kiourtis, A. Mavrogiorgou, K. Mavrogiorgos, D. Kyriazis, A. Graziani, C. Symvoulidis, G. Bella, S. Bocca, F. Torelli, Electronic health records at People's hands across Europe: The InteropEHRate protocols, Stud. Health Technol. Inform. 299 (2022) 145–150, http://dx.doi.org/10.3233/SHTI220973.

[11] O. Borgogno, G. Colangelo, Data sharing and interoperability: Fostering innovation and competition through APIs, Comput. Law Secur. Rev. (ISSN: 0267-3649) 35 (5) (2019) 105314, http://dx.doi.org/10.1016/j.clsr.2019.03.008.

[12] A.U. Otokiti, M.M. Ozoude, K.S. Williams, R.A. Sadiq-onilenla, S.A. Ojo, L.B. Wasarme, S. Walsh, M. Edomwande, The need to prioritize model-updating processes in clinical artificial intelligence (AI) models: Protocol for a scoping review, JMIR Res. Protoc. (ISSN: 1929-0748) 12 (2023) e37685, http://dx.doi.org/10.2196/37685.

[13] D.A. Jenkins, G.P. Martin, M. Sperrin, R.D. Riley, T.P. Debray, G.S. Collins, N. Peek, Continual updating and monitoring of clinical prediction models: time for dynamic prediction systems? Diagn. Progn. Res. 5 (2021) 1–7.

[14] F. Bayram, B.S. Ahmed, a. Kassler, From concept drift to model degradation: An overview on performance-aware drift detectors, Knowl.-Based Syst. (ISSN: 0950-7051) 245 (2022) 108632, http://dx.doi.org/10.1016/j.knosys.2022.108632.

[15] R.I. Horwitz, M.E. Charlson, B.H. Singer, Medicine based evidence and personalized care of patients, Eur. J. Clin. Invest. 48 (7) (2018) e12945.

[16] K.D. Mandl, D. Gottlieb, J.C. Mandel, V. Ignatov, R. Sayeed, G. Grieve, J. Jones, A. Ellis, A. Culbertson, Push button population health: The SMART/HL7 FHIR bulk data access application programming interface, npj Digit. Med. 3 (2020) http://dx.doi.org/10.1038/s41746-020-00358-4.

[17] D. Baylor, K. Haas, K. Katsiapis, S. Leong, R. Liu, C. Menwald, H. Miao, N. Polyzotis, M. Trott, M. Zinkevich, Continuous training for production ML in the TensorFlow extended (TFX) platform, in: 2019 USENIX Conference on Operational Machine Learning (OpML 19), USENIX Association, Santa Clara, CA, ISBN: 978-1-939133-00-7, 2019, pp. 51–53, Available from: https://www.usenix.org/conference/opml19/presentation/baylor.

[18] I. Prapas, B. Derakhshan, A.R. Mahdiraji, V. Markl, Continuous training and deployment of deep learning models, Datenbank-Spektrum (2021) http://dx.doi.org/10.1007/s13222-021-00386-8.

[19] R. Hadsell, D. Rao, a.A. Rusu, R. Pascanu, Embracing change: Continual learning in deep neural networks, Trends in Cognitive Sciences 24 (2020) 1028–1040, http://dx.doi.org/10.1016/J.TICS.2020.09.004.

[20] Z. Zhao, A. Chen, W. Hou, J.M. Graham, H. Li, P.S. Richman, H.C. Thode, A.J. Singer, T.Q. Duong, Prediction model and risk scores of ICU admission and mortality in COVID-19, PLoS One 15 (7) (2020) 1–14, http://dx.doi.org/10.1371/journal.pone.0236618.

[21] V.N. Prachand, R. Milner, P. Angelos, M.C. Posner, J.J. Fung, N. Agrawal, V. Jeevanandam, J.B. Matthews, Medically necessary, time-sensitive procedures: Scoring system to ethically and efficiently manage resource scarcity and provider risk during the COVID-19 pandemic, J. Am. Coll. Surg. (ISSN: 1072-7515) 231 (2) (2020) 281–288, http://dx.doi.org/10.1016/j.jamcollsurg.2020.04.011.

[22] a.K. Triantafyllidis, A. Tsanas, Applications of machine learning in real-life digital health interventions: Review of the literature, J. Med. Internet Res. (ISSN: 1438-8871) 21 (4) (2019) e12286, http://dx.doi.org/10.2196/12286.

[23] S.-C. Huang, A. Pareek, S. Seyyedi, I. Banerjee, M.P. Lungren, Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines, NPJ Digit. Med. 3 (1) (2020) 136.

[24] P. Rajpurkar, E. Chen, O. Banerjee, E.J. Topol, AI in health and medicine, Nat. Med. 28 (1) (2022) 31–38.

[25] Y.-J. Tseng, C.-E. Huang, C.-N. Wen, P.-Y. Lai, M.-H. Wu, Y.-C. Sun, H.-Y. Wang, J.-J. Lu, Predicting breast cancer metastasis by using serum biomarkers and clinicopathological data with machine learning technologies, Int. J. Med. Inform. (ISSN: 1386-5056) 128 (2019) 79–86, http://dx.doi.org/10.1016/j.ijmedinf.2019.05.003.

[26] Y.-J. Tseng, H.-Y. Wang, T.-W. Lin, J.-J. Lu, C.-H. Hsieh, C.-T. Liao, Development of a Machine Learning Model for Survival Risk Stratification of Patients With Advanced Oral Cancer, JAMA Netw. Open (ISSN: 2574-3805) 3 (8) (2020) e2011768, http://dx.doi.org/10.1001/jamanetworkopen.2020.11768.

[27] T.J. Osterman, M. Terry, R.S. Miller, Improving cancer data interoperability: The promise of the minimal common oncology data elements (mCODE) initiative, JCO Clin. Cancer Inform. (2020) 993–1001, http://dx.doi.org/10.1200/cci.20.00059.

[28] A.C. Griffin, L. He, A.P. Sunjaya, a.J. King, Z. Khan, M. Nwadiugwu, B. Douthit, V. Subbian, V. Nguyen, M. Braunstein, C. Jaffe, T. Schleyer, Clinical, technical, and implementation characteristics of real-world health applications using FHIR, JAMIA Open (ISSN: 2574-2531) 5 (4) (2022) ooac077, http://dx.doi.org/10.1093/jamiaopen/ooac077.

[29] G.E. Soto, J.A. Spertus, Epoch® and eprism®: A web-based translational framework for bridging outcomes research and clinical practice, Comput. Cardiol. 34 (2007) 205–208, http://dx.doi.org/10.1109/CIC.2007.4745457,

[30] M. Khalilia, M. Choi, A. Henderson, S. Iyengar, M. Braunstein, J. Sun, Clinical predictive modeling development and deployment through FHIR web services, in: AMIA Annu. Symp. Proc., Vol. 2015, 2015, pp. 717–726.

[31] J. Gruendner, T. Schwachhofer, P. Sippl, N. Wolf, M. Erpenbeck, C. Gulden, L.A. Kapsner, J. Zierk, S. Mate, M. Stürzl, R. Croner, H.-U. Prokosch, D. Toddenroth, KETOS: Clinical decision support and machine learning as a service – A training and deployment platform based on docker, OMOP-CDM, and FHIR Web Services, PLoS One 14 (10) (2019) 1–16, http://dx.doi.org/10.1371/journal.pone.0223010.

[32] D. Bender, K. Sartipi, HL7 FHIR: An Agile and RESTful approach to healthcare information exchange, in: Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems, 2013, pp. 326–331, http://dx.doi.org/10.1109/CBMS.2013.6627810.

[33] A. Vaidyam, J. Halamka, J. Torous, Enabling research and clinical use of patient-generated health data (the mindLAMP Platform): Digital phenotyping study, JMIR Mhealth Uhealth (ISSN: 2291-5222) 10 (1) (2022) e30557, http://dx.doi.org/10.2196/30557.

[34] S. El-Sappagh, F. Ali, A. Hendawi, J.H. Jang, K.S. Kwak, A mobile health monitoring-and-treatment system based on integration of the SSN sensor ontology and the HL7 FHIR standard, BMC Med. Inform. Decis. Mak. 19 (2019) 1–36, http://dx.doi.org/10.1186/S12911-019-0806-Z/FIGURES/11.

[35] J. Gruendner, T. Schwachhofer, P. Sippl, N. Wolf, M. Erpenbeck, C. Gulden, L.A. Kapsner, J. Zierk, S. Mate, M. Stürzl, R. Croner, H.U. Prokosch, D. Toddenroth, Ketos: Clinical decision support and machine learning as a service – A training and deployment platform based on Docker, OMOP-CDM, and FHIR Web Services, PLoS One 14 (2019) http://dx.doi.org/10.1371/journal.pone.0223010.

[36] M. Ayaz, M.F. Pasha, M.Y. Alzahrani, R. Budiarto, D. Stiawan, The Fast Health Interoperability Resources (FHIR) standard: systematic literature review of implementations, applications, challenges and opportunities, JMIR Med. Inform. 9 (7) (2021) e21929.

[37] A. Mavrogiorgou, A. Kiourtis, M. Touloupou, E. Kapassa, D. Kyriazis, Internet of medical things (IoMT): acquiring and transforming data into HL7 FHIR through 5G network slicing, Emerg. Sci. J. 3 (2) (2019) 64–77.

[38] K.L. Morgan, P.V. Kukhareva, P.B. Warner, J. Wilkof, M. Snyder, D. Horton, T. Madsen, J. Habboushe, K. Kawamoto, Using CDS hooks to increase SMART on FHIR app utilization: a cluster-randomized trial, J. Am. Med. Inform. Assoc. 29 (2022) 1461–1470, http://dx.doi.org/10.1093/jamia/ocac085.

[39] HL7, Boston Children's Hospital, Cds hooks, 2022, Available from: https://cds-hooks.hl7.org/.

[40] J.C. Mandel, D.A. Kreda, K.D. Mandl, I.S. Kohane, R.B. Ramoni, SMART on FHIR: A standards-based, interoperable apps platform for electronic health records, J. Am. Med. Inform. Assoc. 23 (2016) 899–908, http://dx.doi.org/10.1093/jamia/ocv189.

[41] L.-C. Kuo, T.-Y. Ting, C.-Y. Lien, Taiwan cancer registry long form implementation guide, 0.1.0 - ci-build, 2023, https://mitw.dicom.org.tw/IG/TWCR_LF/index.html. (Accessed 10 December 2023).

[42] D. Kreuzberger, N. Kühl, S. Hirschl, Machine learning operations (MLOps): Overview, definition, and architecture, IEEE Access 11 (2023) 31866–31879, http://dx.doi.org/10.1109/ACCESS.2023.3262138.

[43] J.-F. Hong, Y.-J. Tseng, Performance vs. Privacy: Evaluating the performance of predicting second primary cancer in lung cancer survivors with privacy-preserving approaches, in: 2022 IEEE-EMBS International Conference on Biomedical and Health Informatics, BHI, 2022, pp. 1–4, http://dx.doi.org/10.1109/BHI56158.2022.9926935.

[44] K.F. Chen, S.H. Liu, C.H. Li, C.C. Wu, C.H. Chaou, I.S. Tzeng, Y.H. Hsieh, G.N. Blaney, Z.Y. Liu, S.T. Han, Y.L. Chan, Development and validation of a parsimonious and pragmatic CHARM score to predict mortality in patients with suspected sepsis, Am. J. Emerg. Med. 35 (2017) 640–646, http://dx.doi.org/10.1016/j.ajem.2016.10.075.

[45] HL7 International - Vocabulary Work Group, HL7.terminology - ActEncounterCode - FHIR v4.0.1, 2022, Available from: https://terminology.hl7.org/3.1.0/ValueSet-v3-ActEncounterCode.html.

[46] A.D. Haimovich, N.G. Ravindra, S. Stoytchev, H.P. Young, F.P. Wilson, D. van Dijk, W.L. Schulz, R. andrew Taylor, Development and validation of the quick COVID-19 severity index: A prognostic tool for early clinical decompensation, Ann. Emerg. Med. 76 (2020) 442–453, http://dx.doi.org/10.1016/j.annemergmed.2020.07.022.

[47] C.-Y. Wu, Analysis of infection risk in necrotizing soft tissue of the extremities using machine learning techniques, 2022, Available from: https://thesis.lib.cgu.edu.tw/thesis/detail/1898ab0bf00a94298dfd6a3a6f843442/.

[48] M.-S. Tsai, M.-H. Lin, C.-P. Lee, Y.-H. Yang, W.-C. Chen, G.-H. Chang, Y.-T. Tsai, P.-C. Chen, Y.-H. Tsai, Chang Gung Research Database: A multi-institutional database consisting of original medical records, Biomed. J. 40 (5) (2017) 263–269.

[49] HL7.org, Observation - FHIR v4.0.1, 2019, Available from: http://hl7.org/fhir/R4/observation-examples.html.

[50] Z.-M. Kuo, GitHub - MoCab, the models, 2023, Available from: https://github.com/DHLab-TSENG/MoCab/tree/main/Backend/mocab_models.

[51] K. Kawamoto, P.V. Kukhareva, C. Weir, M.C. Flynn, C.J. Nanjo, D.K. Martin, P.B. Warner, D.E. Shields, S. Rodriguez-Loya, R.L. Bradshaw, et al., Establishing a multidisciplinary initiative for interoperable electronic health record innovations at an academic medical center, JAMIA Open 4 (3) (2021) ooab041.

[52] Smile CDR, HAPI FHIR - The open source FHIR API for Java, 2022, Available from: https://hapifhir.io/.

[53] GitHub - nazrulworld/fhir.resources: FHIR resources, 2023, https://www.hl7.org/fhir/resourcelist.html. Available from: https://github.com/nazrulworld/fhir.resources.

[54] HL7.org, StructureDefinition: VitalSigns - FHIR v4.0.1, 2019, Available from: https://hl7.org/fhir/R4/vitalsigns.html.

[55] HL7.org, ImplementationGuide - FHIR v5.0.0-cibuild, 2023, Available from: https://build.fhir.org/implementationguide.html.

[56] Y. Gupta, M. Chhetry, K.R. Pathak, R.K. Jha, N. Ghimire, N.K.M.B. Nath, Karn, G.K. Singh, J.N. Bhagabati, Risk factors for necrotizing fasciitis and its outcome at A tertiary care centre, J. Ayub Med. Coll. Abbottabad 28 (2016) 680–682.

[57] L.B. Travis, The epidemiology of second primary cancers, Cancer Epidemiol. Biomark. Prev. 15 (11) (2006) 2020–2026.

[58] J.V. Aredo, S.J. Luo, R.M. Gardner, N. Sanyal, E. Choi, T.P. Hickey, T.L. Riley, W.-Y. Huang, A.W. Kurian, A.N. Leung, et al., Tobacco smoking and risk of second primary lung cancer, J. Thorac. Oncol. 16 (6) (2021) 968–979.

[59] K.D. Lee, S.C. Chen, C.H. Chan, C.H. Lu, C.G. Chen, J.T. Lin, M.F. Chen, S.H. Huang, C.M. Yeh, M.C. Chen, Increased risk for second primary malignancies in women with breast cancer diagnosed at young age: a population-based study in Taiwan, Cancer Epidemiol. Biomark. Prev. 17 (2008) 2647–2655, http://dx.doi.org/10.1158/1055-9965.EPI-08-0109.

[60] K.C. Thandra, A. Barsouk, K. Saginala, J.S. Aluru, A. Barsouk, Epidemiology of lung cancer, Contemp. Oncol./Współczesna Onkol. 25 (1) (2021) 45–52.

[61] E. Choi, N. Sanyal, V.Y. Ding, R.M. Gardner, J.V. Aredo, J. Lee, J.T. Wu, T.P. Hickey, B. Barrett, T.L. Riley, et al., Development and validation of a risk prediction model for second primary lung cancer, JNCI: J. Natl. Cancer Inst. 114 (1) (2022) 87–96.

[62] HL7.fhir.US.mcode - FHIR v4.0.1, 2023, Available from: https://build.fhir.org/ig/HL7/fhir-mCODE-ig/index.html.

[63] cancer2fhir, Gazelle test for Cancer Registry - HackMD, 2021, Available from: https://hackmd.io/@cancer2fhir/BJSmqeOrK.

[64] SMART on FHIR, GitHub - smart-on-fhir/smart-dev-sandbox: Docker based sandbox for smart apps, 2023, Available from: https://github.com/smart-on-fhir/smart-dev-sandbox.

[65] M.T. Majeed, Information and communication technology (ICT) and environmental sustainability in developed and developing countries, Pak. J. Commer. Soc. Sci. 12 (3) (2018) 758–783.

[66] D.A. Higón, R. Gholami, F. Shirazi, ICT and environmental sustainability: A global perspective, Telemat. Inform. 34 (4) (2017) 85–95.

[67] J.-R. Yu, C.-H. Chen, T.-W. Huang, J.-J. Lu, C.-R. Chung, T.-W. Lin, M.-H. Wu, Y.-J. Tseng, H.-Y. Wang, Energy efficiency of inference algorithms for clinical laboratory data sets: Green artificial intelligence study, J. Med. Internet Res. (ISSN: 1438-8871) 24 (1) (2022) e28036, http://dx.doi.org/10.2196/28036, Available from: https://www.jmir.org/2022/1/e28036.

[68] F.N. Khan, A. Sana, U. Arif, Information and communication technology (ICT) and environmental sustainability: a panel data analysis, Environ. Sci. Pollut. Res. 27 (2020) 36718–36731.

[69] a. Karabetian, A. Kiourtis, K. Voulgaris, P. Karamolegkos, Y. Poulakis, A. Mavrogiorgou, D. Kyriazis, An environmentally-sustainable dimensioning workbench towards dynamic resource allocation in cloud-computing environments, in: 2022 13th International Conference on Information, Intelligence, Systems & Applications, IISA, IEEE, 2022, pp. 1–4.

[70] R.M. Al-Tam, A.M. Al-Hejri, S.M. Narangale, N.A. Samee, N.F. Mahmoud, M.A. Al-masni, M.A. Al-antari, A hybrid workflow of residual convolutional transformer encoder for breast cancer classification using digital X-ray mammograms, Biomedicines 10 (2022) 2971, http://dx.doi.org/10.3390/BIOMEDICINES10112971, 2022, Vol. 10, Page 2971.

[71] J.H. Lee, G.W. Lee, G. Bong, H.J. Yoo, H.K. Kim, Deep-learning-based detection of infants with autism spectrum disorder using auto-encoder feature representation, Sensors 20 (2020) 6762, http://dx.doi.org/10.3390/S20236762, 2020, Vol. 20, Page 6762.

[72] HL7.org, Clinical quality language (CQL) — cql.hl7.org, 2023, https://cql.hl7.org/. (Accessed 10 December 2023).