Mass General Brigham

NVIDIA

Large-Scale Model Validation in Healthcare

White Paper

August 2022

Authors

Christopher P. Bridge[1*]
Alma Fredriksson[1*]
Jiahui Guan[2]
Sidney Bryson[2]
James M. Hillis[1]
Sarah Mercaldo[1]
Ryan J. Morley[1]
Matthew D. Li[1]
Xiang Li[1]
Eric L'Italien[1]
Darren Sack[1]
Aoxiao Zhong[1]
Keith J. Dreyer[1]
Mona Flores[2]
Jayashree Kalpathy-Cramer[1]
Quanzheng Li[1]
Leslie R. Lamb[1]
Constance D. Lehman[1]
Thomas Schultz[1]
Katherine P. Andriole[1]
Colin Compas[2]
Bernardo C. Bizzo[1†]
Risto Haukioja[2†]

[1]Mass General Brigham, Boston, MA
[2]NVIDIA Corporation, Santa Clara, CA

*The following authors contributed equally to this work.
†The following authors are co-senior authors.

Table of Contents

Introduction

Following the groundbreaking growth of imaging-based artificial intelligence (AI) in healthcare over the last decade, an unprecedented number of AI models are being developed across a multitude of medical subfields, including radiology, cardiovascular disease, and neurology. In 2020 alone, 51 new imaging-based AI and machine learning-enabled medical devices were cleared or approved by the U.S. Food & Drug Administration (FDA) [1]. This exciting trend reflects a growing industry-wide understanding of the enormous potential of AI applications in clinical settings, as well as increased technological capacity and improved access to computational resources. Graphics Processing Units (GPUs), a key part of modern AI infrastructure, dramatically reduce the AI training and inference time. The AI accelerator ecosystem using GPUs provides a framework and toolkits that are optimized specifically for deep learning, which expedite AI development. An additional driver is the availability of large visual databases such as ImageNet, which contains 14 million publicly available annotated images and has enabled the creation of pre-trained deep learning models that can be used for transfer learning in medical imaging [2]. These advances have contributed to a surge in the number of developed medical imaging models, though many are ultimately not used in practice, owing to limited clinical relevance or suboptimal performance in the clinical setting commonly due to poorly designed training data cohorts.

Integration of AI algorithms into clinical workflows requires rigorous model building and external validation, as well as a robust inference pipeline to enable timely results delivery for urgent findings. For the remainder of this white paper, we refer to validation as the evaluation of the predictive performance of an existing prediction model on separate data *after* completion of model development, and distinct from any testing that occurs *during* model development [3]. Guidance on evidence generation during model development can be found in existing literature, such as the framework '*Generating Evidence for Artificial Intelligence-based Medical Devices*' provided by the World Health Organization [4]. Datasets currently used for algorithm development and testing are typically small, as larger sample sizes often require proportional additional efforts, including further cohort and image selection, and expert annotation, which can be expensive, time consuming, and tedious. Research groups often move quickly and the trade-off with producing algorithms mainly focused on research publications sometimes means that ensuring sufficient robustness of tools built for internal or external clinical deployment becomes secondary. Once the final model is deployed clinically, the limited data representation used in model training risks model performance that may be drastically lower than results achieved during model development and initial testing.

External validation with new, independent, data representative of the target population using pre-defined and clinically meaningful performance metrics is critical before clinical use of an algorithm. This practice helps ensure that model predictions are, at least initially, reliable, accurate, and can be used as a tool to assist with clinical decisions while subject to ongoing performance monitoring. Overall, this process allows the identification of potential failure modes on specific variables, such as lack of generalization to device manufacturers or models, exam protocols, technical parameters, or patient subpopulations especially when not used in the model training set. The results from these analyses may also allow for algorithm finetuning, retraining, or set parameters for its clinical use to data in which its performance is acceptable. Validation requirements directly depend on the model design, intended use and clinical context. In supervised imaging-based machine learning, annotation is required and may involve subject matter experts performing classification tasks at the study, series, or image level (e.g., label the presence or absence of a finding), drawing pixel-level mark-ups such as measurements, bounding boxes, segmentation, or a combination of both. Typically, the validation of imaging-based models requires careful ground truth (also known as a reference standard or label) established by experts such as radiologists or technologists using a well-curated dataset on tens to hundreds of cases. However, use cases where the ground truth is based on electronic medical record (EMR) data may be well-suited for large-scale validation on many thousands of cases.

This white paper provides best practices for performing clinically meaningful validation of medical imaging-based AI models and practical model integration into existing radiology environments. It includes guidance on study design, selection of appropriate validation data, creation of an efficient pipeline and utilization of batch inference when appropriate. We conclude with a description of specific use cases where this pipeline has been successfully implemented and an analysis of potential future directions related to AI model clinical validation.

Challenge #1 - Building a generalizable model

Processing large amounts of data comes with technical challenges. Due to data security and patient privacy concerns, many research labs developing AI models are constrained to using a validation dataset gathered at a single hospital site. This dataset is frequently limited to a single or very few acquisition protocols, as additional exam codes may include different or additional technical parameters to account for and additional series, the selection of which is typically conducted manually and requires medical expertise (see *Best Practices – Performing ground truthing at large scale* for further detail). A recent review of 118 FDA-cleared AI/ML algorithms showed that only 56% of them had stated patient data for validation studies and merely 11 algorithms were validated on a dataset consisting of at least 1000 patients [5]. A narrowly validated model is at high risk of generalizing poorly to unseen data and results may not be meaningful in real clinical settings. This applies whether the model is to be deployed locally at the institution where it was developed or is an FDA-cleared device available commercially through marketplaces. As an example, an AI model for cervical spine fracture detection cleared by the FDA in 2019 was subsequently validated at an external hospital site and the model sensitivity was found to be 54.9% (95% CI, 45.7%–63.9%) [6]. If a model is intended for clinical use, safe and meaningful model performance that enhances physician workflows and positively impacts patient care must be ensured.

Validation on large, diverse datasets offers an opportunity to confirm that the model performs well beyond the environment in which it was developed. By employing large-scale batch inference throughout the development process and using validation data representative of the population where it is to be deployed, the chance improves of achieving equivalent results in the final clinical setting as was seen in testing at the end of the model development phase. We hope that the guidance provided in this white paper can help address many of the technical challenges of large-scale validation and make it more widely attainable.

Challenge #2 - Meeting regulatory requirements

A key requirement posed by the FDA is to demonstrate that the model has been designed with careful consideration of the target population. A well-designed model development process includes a deliberate choice of whether the target population comprises all patients with the medical condition of interest or if its intended use is restricted to a specific subgroup or care setting, such as emergency department patients or inpatients. The distribution of the medical condition of interest may differ greatly across these populations and validating a model on representative data to answer the specific clinical need provides better insight into its effective performance.

Large-scale validation using batch inference offers an opportunity to account for subgroup and data diversity regulations early in the process. Knowledge of how the model performs relative to specific thresholds can help with framing a future regulatory submission and determining an appropriate submission class. A Computer Aided Triage (CADt) device may provide clinical support with identification of patients in need of urgent care for especially critical conditions, while a Computer Aided Detection (CADe) device is more appropriate for models that detect and demonstrate the position or size of potential abnormalities in medical images [7]. To ensure sufficient data representation, we recommend moving beyond a validation cohort based on specific exam codes alone to account for different scanners, patient demographics and protocols (see *Best Practices – Selecting appropriate validation data*). Considering diversity measures early and employing an efficient validation pipeline is likely to reduce time to both regulatory submission and approval.

Challenge #3 - Supporting both retrospective and prospective validation

We distinguish between two common workflows for model validation, which will be referred to as *prospective* and *retrospective*. In *retrospective* (or *batch*) validation settings, the model inference is initiated manually on a (typically large) batch of previously acquired studies. Retrospective validation can leverage large historical datasets to quickly provide insights about model performance. However, its applicability may be limited as imaging protocols and imaging equipment change over time. In *prospective validation*, model inference is initiated whenever a suitable study is acquired at the institution as soon as the study becomes available. Thus, depending on the specific intended use, prospective validation may provide a better indication of the likely future performance of the model when deployed into clinical care in a similar setting.

Furthermore, depending on the intended use and claim of benefits of the model, the true value of an AI model may be determined not simply by its performance metrics but by the way in which clinicians and other healthcare professionals interact with it within a clinical workflow. Prospective validation combined with well-designed user studies facilitates estimation of the true impact of the model on clinical care, e. g. through measuring the reduction in interpretation time, the accuracy of a radiologist aided by the model or downstream effects on patient care. It is expected for a model to initially undergo a retrospective validation process, and then, if successful, a subsequent prospective validation process may occur. From a technical perspective, it is desirable to support these two workflows within a single system to reduce duplicated effort and improve efficiency (see *Architecture Design Reference*).

Traditional Imaging Model Validation Pipeline

In this section, we describe the way imaging-based AI models are typically validated in a research and development setting. In such a setting, the responsibility for model validation often falls upon the single data scientist, or small team of data scientists, that originally developed the model. The methodology and tools used will often differ according to the data scientist's preferences, and will often involve development of custom, one-off validation scripts. Complementing the pipeline outlined below is ground truthing, which is described in further detail in *Best Practices – Performing ground truthing at large scale*.

Data Pull, De-Identification, and Data Storage: The images belonging to the selected studies are pulled down to a local or shared filesystem accessible to the data scientist, and may be de-identified in the process according to institutional policies.

Series Selection: Selection of the appropriate imaging series (or multiple series) for input to the model is then conducted, usually by either a custom script or by manual image review. Please see *Best Practices – Selecting appropriate validation dat*a for further guidance.

Model Execution: The model is typically executed by a custom validation program, which processes all specified cases by reading the relevant files directly from a local or shared filesystem using the native model software framework such as *Tensorflow* or *Pytorch*. The hardware utilized is typically a GPU-enabled on premises or cloud-hosted server made available to the data scientist for their experiments.

Model Output Visualization and Analysis: The output of the model will usually be placed into a local or shared filesystem by the model. It may include text files, such as CSV or Javascript Object Notation (JSON) files, containing results, and/or other artifacts such as segmentation masks or heatmaps in a variety of standard and non-standard formats. Subsequently, the results may be visualized in a systematic or ad-hoc manner, and results files may be analyzed to produce summary statistics and identify shortcomings of the model.

The advantages of such a pipeline are that it has few requirements on infrastructure beyond what the data scientist will typically already have access to for model development, and requires little support from other team members such as software engineers and systems administrators. However, it is also far removed from a realistic deployment scenario in which the model sits within a clinical workflow and an enterprise radiology environment. Thus, while this design may provide a reasonable test of the accuracy of the model (if the cohort is suitably representative), it falls short of evaluating the readiness of the model for real-world use. Furthermore, the lack of standardization in this process means that it does not scale well to an institution looking to validate multiple models from different internal and external model developers efficiently using a shared pool of compute resources.

Identifying Tools to Support Large-Scale Inference

Large-scale model validation places heavy demands on enabling technological infrastructure. In this section, we describe tools that we have found valuable in our model validation work, but we recognize that every site and validation team have their own specific needs and requirements. Mass General Brigham uses a OneView platform consolidating medical imaging exams across the entire enterprise into one common platform consisting of modality routing, diagnostic viewing, radiology workflow management, dictation system, long-term archiving, and clinical viewing via the Epic electronic health record system. Modality routing and the vendor neutral archive (VNA) are the two components of the platform that contribute to the effort to test, validate, and deploy models. NVIDIA Clara [NVIDIA Corporation, Santa Clara, CA] provides an AI platform that manages all the AI inferences and optimization, compute resource allocations and AI pipelines' maintenance.

Mass General Brigham performs modality routing through the Laurel Bridge Compass Routing Workflow Manager [Laurel Bridge Software, Inc., Newark, DE], using an infrastructure of 16 Compass servers that ingest every exam from the nearly 1600 devices across the enterprise. Through a built-in rules interface, Compass delivers exams to various destinations based on certain conditions and criteria. Custom scripting through Compass interface is also possible, allowing for a variety of use cases. For example, custom scripting can be installed for writing DICOM data out to a database or picking up data and storing to the DICOM header. Compass also allows for de-identification of exams and tag-morphing in flight, with the ability to update various tags based on specific conditions. All exams that come through Compass are stored in the GE VNA [GE Healthcare, Chicago, IL], which is a long-term archive. The historical data at our member institutions have also been migrated to the VNA for retention.

These two applications play an important role in model testing and validation. As the central point for all data flow, Compass is primed to play the role of a traffic police officer. Creating and adjusting routing rules based on the image selection criteria of the model is easy to accomplish in the Compass user interface. Advanced routing logic can easily be achieved through the custom capabilities of Compass with simple C# scripting. The user interface and custom scripting in Compass allow for flexibility to provide the correct imaging needed for the model. With the tag morphing functionality, Compass can store any other data type required in any of the DICOM tags that can be read by the model. While crude, this can be useful as a simple means to communicate the results of selection processes to the machine learning model. Other methods in compliance with existing standards, such as a DICOM Key Object Selection (KOS) document may provide better ways to communicate this information.

The GE VNA, by nature of its job as our archive of record, houses all medical imaging data performed at our institutions. Any type of imaging data that may be needed can be retrieved from the archives for training and testing models. The VNA is also able to accept DICOMWeb requests and move data and exams based on QIDO/WADO calls. Mass General Brigham also has a separate instance of the VNA that is used primarily for research that can house de-identified copies of the clinical data. The research teams can manipulate this copied data without compromising the historical clinical record.

In addition to Compass and the GE VNA, Mass General Brigham uses the AI platform NVIDIA Clara to bring deep learning models trained and validated on thousands of retrospective or prospective studies from research labs into the clinical workflow. A growing number of machine learning engineers are beginning to package AI models into containers; however, these have previously not been practically deployed in a clinical system that includes integration with existing hospital IT systems. NVIDIA Clara provides this, along with a container-orchestration system for automating AI application deployment, with its container-based, cloud-native development and deployment framework. It uses Kubernetes under the hood and allows researchers or developers to break down complicated AI workflows into multi-staged container-based pipelines. NVIDIA Clara also comes with other microservices such as workload management, AI inference accelerator Triton, monitoring, profiling and 3D visualization (see *Architecture Design Reference* for further details).

The clinical environment of Compass and VNA allows for not only access to reams of data, but these tools also allow customization of the data to make it as compatible with the requirements of the model as possible. The rules engine of Compass provides the ability to hone down to the exact image that is needed for inference. It allows models to be tested against live clinical data and also allows for easy integration into the clinical service once validation is completed. This is complemented by the Kubernetes-based NVIDIA Clara, which provides a standalone AI platform that enables fast inference performance, handles models in a scalable way and allocates resources efficiently.

Architecture Design Reference

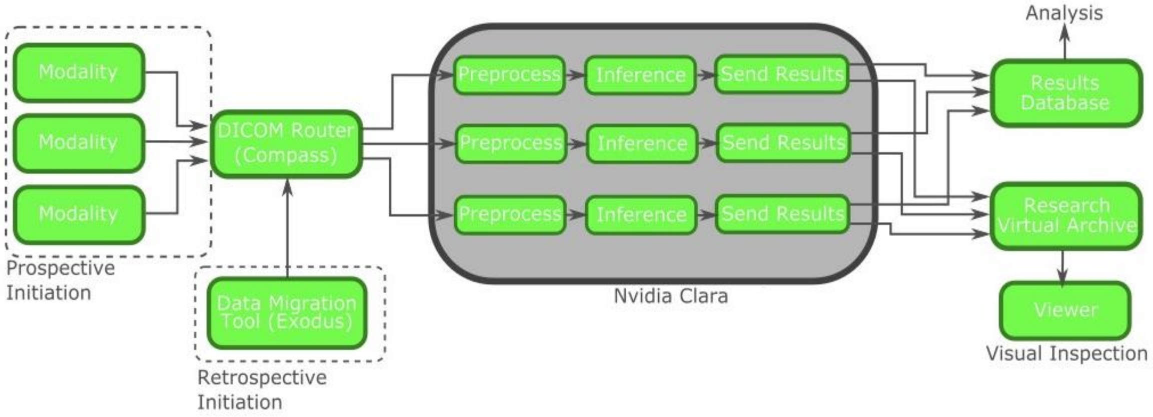


Figure 1. Overview of the model validation system

An overview of our platform for large scale model validation is presented in *Fig. 1*. The system consists of 4 major components:

1. **DICOM Router** – The DICOM routing software is set up to receive DICOM studies acquired from a large number of imaging modalities located in the hospitals, including CT, MRI, ultrasound, x-ray, and mammography imaging devices, as part of infrastructure to support the hospitals' clinical radiology operations. We additionally leverage the functionality of this software for the purposes of AI imaging model validation. For each AI model undergoing validation, the DICOM routing software is configured to select appropriate DICOM studies and series from clinical imaging and forward them on to the NVIDIA Clara platform for processing.

2. **NVIDIA Clara**– The AI models are packaged to run within NVIDIA Clara platform as NVIDIA Clara operators, when invoked by an incoming study from the DICOM router. Each model in the platform is represented by a multi-stage pipeline that runs on the incoming input study, performs GPU- or CPU-enabled model inference, and creates outputs. NVIDIA Clara supports building and reusing operators that take the output and deliver them in the form of textual or numerical values and DICOM objects, which are respectively sent on to the results database or the research virtual archive.

3. **AI Results Database** – A SQL database is used to store numerical and textual outputs from the AI model to enable later analysis by investigators.

4. **Research Virtual Archive** – A virtual DICOM archive is set up to receive AI model results that are contained within DICOM-format objects, such as segmentations, structured reports, and secondary captures.

In this section, we give further details about the setup and configuration of these components.

Prospective vs. Retrospective Initiation

Both retrospective and prospective validation are supported by this system. In the *prospective* scenario, the DICOM routing software is configured to identify studies that are sent through the routing software as they are acquired from modalities. In *retrospective* (or *batch*) initiation, the model inference is initiated manually on a (typically large) batch of previously acquired studies that already reside in the clinical VNA. In this scenario, a further data migration component is required. In our institution, Laurel Bridge's Exodus migration utility serves this purpose. Exodus is configured to send a batch of pre-selected studies to Compass from the institution's clinical VNA, and Compass then routes these studies to the AI model. The data migration tool may be needed to reduce the rate at which it sends images in order to avoid overwhelming the resources of NVIDIA Clara.

Aside from the use of the data migration tool, and configuration of the data source on the DICOM router, the rest of the system configuration remains the same for both workflows. The advantage of this approach is that a retrospective analysis may closely mimic a prospective analysis, meaning that it can provide a more realistic assessment of model readiness for clinical integration. Furthermore, it is straightforward to move a model that is configured for retrospective initiation into a prospective workflow if and when it is deemed ready.

DICOM Router Configuration

The purpose of the DICOM routing software is to select images appropriate for each deployed AI model from the studies acquired for clinical care and forward them on to NVIDIA Clara for processing. While the criteria used will be similar between different routing software, their implementation will vary.

Within Laurel Bridge's Compass, a new rule is established in the routing configuration for each AI model undergoing validation. This rule typically has two tasks:

- **Study Selection:** imaging studies are selected based on their exam code, which is a short code that specifies the type of imaging study performed. While definitions of exam code vary by institution, they will typically identify characteristics of the study such as the imaging modality used, the body part examined in the study and the presence of contrast agents used in the study. The DICOM routing software is therefore configured to route studies from one or more exam codes to a particular AI model. For example, this step could be used to select chest x-ray studies, non-contrast abdominal CT studies, or contrast-enhanced brain MR studies for processing by an AI model.

- **Series Selection:** an imaging study will typically contain multiple series of images acquired from the patient during the study. Commonly only one particular series is required as input to the AI model. The DICOM routing software is configured to identify the correct series to use based on the values of particular DICOM attributes of the DICOM objects within a series. In some simple situations, using the "Series Description" attribute may be sufficient, but often further attributes for characteristics such as orientation and position information or imaging acquisition parameters may be required.

The potential complexity of determining appropriate rules should not be underestimated. The identification of different series may not be standardized or easy to determine from DICOM attributes. For example, the identification of sequences of interest within MRI studies can be especially challenging due to the large number of different sequences in use and the lack of a single reliable and standardized mechanism to identify different sequences within the DICOM header. This has led some to develop more elaborate methods based on machine learning to categorize series based on DICOM attributes that are reliably present in the header [8]. Clinical stakeholders, the AI model developer(s), and radiology IT staff should collaborate closely at this stage to ensure their respective considerations and expertise are taken into account.

While it is possible to perform the function of series selection using custom code within the NVIDIA Clara operator for the AI model, we recommend against this approach in order to allow the AI model operator to be agnostic to the environment in which it is deployed. However, in certain more complex situations, it may be necessary to have a NVIDIA Clara operator perform some parts of the series selection task. Once appropriate rules have been determined, the DICOM router is configured to send the appropriate images via the DICOM communications protocol to a model-specific application entity (AE) title set up in the NVIDIA Clara DICOM adaptor.

AI Inference Platform

NVIDIA's AI Inference platform is a critical piece in the whole workflow that facilitates deployment of inference models (see *Fig. 2*). It leverages several existing software products, commonly used to perform inferences, including Docker, Kubernetes, Helm, YAML, and NVIDIA-specific components (NVIDIA Docker Runtime and NVIDIA CUDA). Additionally, the NVIDIA Clara Command-Line Interface (NVIDIA Clara CLI) provides commands for configuration and monitoring of the system.

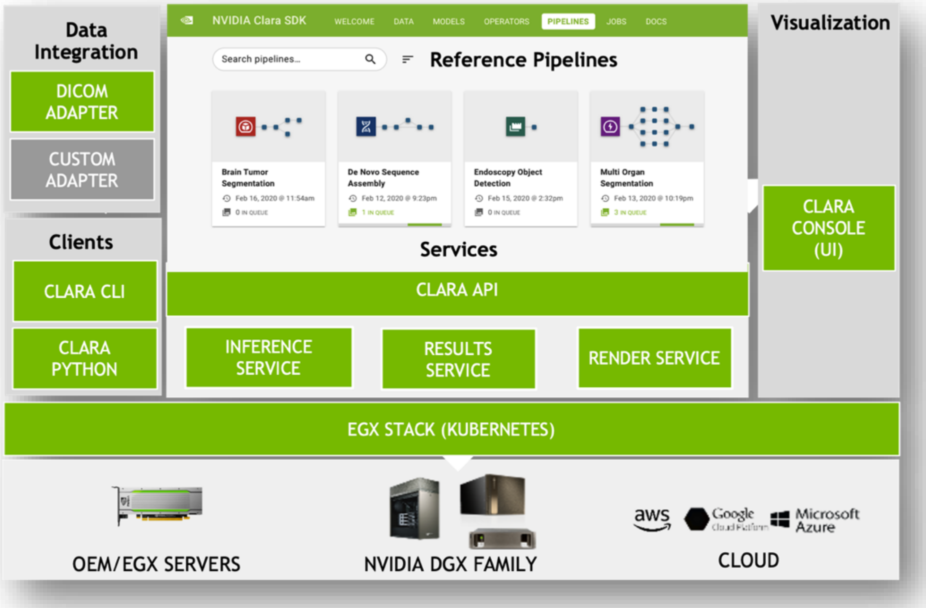


Figure 2. Overview of inference platform

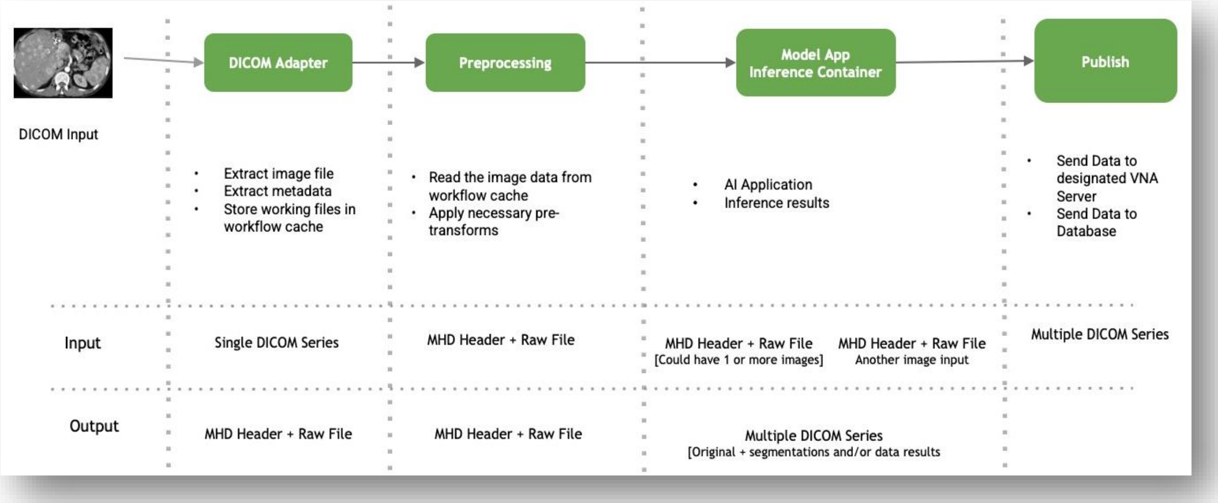
Installation: NVIDIA Clara can be installed using Ansible, a tool for configuration management. General scripts are provided by NVIDIA and should be customized to fit the environment. Specifically, Ansible scripts will download and install the required software, configure the server to run containers, and configure NVIDIA Clara. Users should confirm the versions required by different components, as version changes may cause issues.

Usage: NVIDIA Clara supports a service to interface with hospitals' DICOM routing solutions as an internal operator called DICOM adapter. The DICOM adapter receives DICOM studies as DICOM Service Class Provider (SCP). Each study has an AE title and a DICOM adapter associates the AE title with the corresponding AI pipeline. Therefore, PACS router forwards matching studies to the DICOM adapter using AE title which in turn triggers the AI pipeline. For example, a DICOM client initiates a C-MOVE request to PACS to transfer stored SOP instances to DICOM adapter using C-STORE operations.

Once NVIDIA Clara receives the study via DICOM-adapter, it enables the creation of a pipeline, with which one or more containers (called operators) can be used to process data and perform inference. Pipeline operators can include stages such as: file conversion, e.g. from DICOM to NIfTI, image extraction, preprocessing, inference, result preparation, result distribution, and reporting, as illustrated in *Fig. 3*.

As an example, we may want a pipeline to perform the following:

1. Receive a study via DICOM (C-STORE)
2. Pass the study data to an operator that performs image preprocessing
3. Pass the preprocessed image to another operator for inference
4. Pass inference output to a final operator to be sent to a HTTP web service.



	DICOM Adapter	Preprocessing	Model App Inference Container	Publish
	• Extract image file • Extract metadata • Store working files in workflow cache	• Read the image data from workflow cache • Apply necessary pre-transforms	• AI Application • Inference results	• Send Data to designated VNA Server • Send Data to Database
Input	Single DICOM Series	MHD Header + Raw File	MHD Header + Raw File [Could have 1 or more images] MHD Header + Raw File Another image input	Multiple DICOM Series
Output	MHD Header + Raw File	MHD Header + Raw File	Multiple DICOM Series [Original + segmentations and/or data results	

Figure 3. Example pipeline flow.

Such a pipeline would be configured using a YAML file to define the pipeline and NVIDIA Clara CLI to install. Once the pipeline has been configured, activity can be monitored and troubleshot using command-line tools. Additionally, one can make use of the Platform API using NVIDIA's Python client, which facilitates API access [9]. Further code and details of pipeline configuration, activity monitoring and troubleshooting are available in the *Appendix*.

Packaging an AI Model as an Operator: Creation of an operator implementing the AI model requires building an application container image. The container image when run with a particular (configurable) command, runs the model on data found in its input directory and writes any results into the output directory. Clara provides standard packaging templates that are defined in a YAML file, which helps to generate such app container images. All pre- or post-processing steps can be bundled into a single container or can be separated into multiple containers. Each container is considered an operator and the Clara pipeline combines those operators to bring the AI model into an enterprise-grade AI inference workflow.

We recommend the following steps in preparation for model integration into NVIDIA Clara:

- Refactor the application code to run from input DICOM files (or another format such as NIfTI if an appropriate pre-processing operator is being utilized earlier in the pipeline), and write results out to an output directory, within a single process.
- Build a container image that contains the application code, any further resources required (such as model weights), and all runtime dependencies of the application. The model developer has freedom to install any necessary software packages or include any necessary files in this stage.
- Configure the application container as an operator within NVIDIA Clara by specifying the image name, input and output directory location, and command to execute the inference process.

AI Results Database

The AI results database is a SQL database used to store textual and numerical outputs from the model for later analysis. The model code pipeline implemented by the model developer is required to place a file in its output directory in the JSON format meeting a certain specification and containing any results to be recorded such as the predicted classification label, prediction score, bounding box coordinates or segmented volume. Then, a simple custom "Send Results" operator is placed at the end of the NVIDIA Clara pipeline to read these results and send them to the AI Results Database.

Research Virtual Archive

The research virtual archive (VA) is an archive configured on our GE VNA specifically for the purpose of storing DICOM objects created by models undergoing validation. Models may create DICOM objects in order to store detailed results that are not appropriate for the AI Results Database, to enable visualization of model outputs by investigators and clinical stakeholders, or to provide model output in a format compatible with clinical PACS systems. Some common DICOM-format outputs include:

- **DICOM Segmentations:** Segmentation masks in raster format from segmentation models.
- **DICOM RT (Radiotherapy) Struct:** Segmentation masks in vector format from segmentation models.

- **DICOM Structured Reports (SRs):** SRs can store various types of results, including textual and numerical results and vector graphic objects.
- **DICOM Grayscale Softcopy Presentation States (GSPS):** GSPS objects are used to display model results in the form of text and/or polygons overlaid on the original image.
- **DICOM Secondary Captures:** Secondary captures can store arbitrary generated images in order to visualize model outputs.
- **DICOM Parametric Maps:** Parametric maps may be used to store quantitative maps of image analysis results, such as a saliency map from a classification model.

Please note that although it is straightforward to store any of these DICOM objects created by an AI model in the research VA, viewing the files thereafter depends upon the availability of a DICOM viewer supporting the relevant object. However, NVIDIA Clara does support a rendering service that will take a variety of formats and produce a visual display.

NVIDIA Clara provides pre-built operators to produce DICOM Segmentations and DICOM RT Struct objects. Other DICOM output files may be created using libraries such as *highdicom*. Once generated, NVIDIA Clara provides an existing operator (“register-dicom-output”) that can be used to send the objects to the VA using the DICOM communications protocol.

Latency and throughput considerations

Another important consideration in large-scale model validation is latency and throughput. Having a near real-time inference speed may be critical in a clinical live inference workflow and may also be relevant in retrospective and prospective studies. One way to increase the latency performance is leveraging accelerated AI inference frameworks such as Triton. To scale-out the end-to-end AI workflow, we also need to assess the key performance metrics such as the total number of studies processed, and the throughput and stability of the overall AI system. In this case, throughput is the amount of processing time per study, and stability is how long the system can automatically process studies without administrative or manual intervention. For example, a stability issue of the workflow is created by having the admin constantly monitor disk utilization and manually remove data without completely informing Kubernetes and NVIDIA Clara. To establish an enterprise-grade AI batch workflow for large-scale model testing, we need to optimize the workflow in terms of latency, throughput and stability.

Queuing

Large-scale validation allows for thousands of studies to be submitted as jobs to the AI workflow as fast as possible. However, in practice each job or inference run through the pipeline takes an average amount of time based on the size of the input data set, the speed of the algorithm, network delays, and I/O driven operations with disk, memory and components. The system must be able to manage the queuing of these workloads such that each AI job and its outcome is evaluated for its resource needs, prioritized based on user flags, and tracked from arrival to completion.

Resource Management

While there may be thousands of queued jobs in a large-scale validation workload, the resources available at any given time are usually only able to perform a small number of these complete job pipelines in parallel. For example, if each job requires a GPU with a minimum defined set of GPU memory, and the system only contains four GPU resources with those requirements, then the system is limited to four parallel jobs. NVIDIA Clara is able to evaluate the provided system resources *vs*. resource requirements of each queued job and pre-determine whether the job will have enough resources before starting. If there are multiple resource constraints (e.g. disk space, GPU memory, and CPU memory) NVIDIA Clara can determine when those resources are available and then execute the pipeline to completion. This resource management combined with the queuing capability is critical to enabling an AI workflow that scales when presented with thousands of job requests each with different resource requirements simultaneously.

Stability

Some resources like disk resources are often taken for granted in today's cloud storage with near infinite theoretical capacity. However, in closed systems typically found in hospital IT infrastructure, those resources are extremely finite. Further, medical imaging is relatively large in scale and not always uniform in size depending on series availability and the imaging formats selected. NVIDIA Clara, in addition to compute resource management, has clean-up mechanisms and backoff controls to keep the AI workflow engine from experiencing overload from external systems. If the volume of data in the requested studies from the PACS exceeds the disk queue capacity, then the system protects itself by rejecting new jobs at the router before they are accepted into the system. Typical routing solutions further signal to the source or throttle the requests sent to the AI platform. NVIDIA Clara and the DICOM adapter operator will reject new job requests if the jobs cannot be scheduled with adequate resources. This mechanism provides stability against large scale jobs unintentionally overwhelming the job resources. In addition, NVIDIA Clara pipelines dictate that each operator explicitly declares resource requirements to provide strong checking of resource allocation before starting jobs. This set-up allows the AI system to operate without taking any individual job that would exceed the capacity of the system.

Throughput

AI workflow throughput is often a result of I/O design, model design, and complexity of components in the pipeline. If the most constrained time component of the pipeline is the time it takes to load the image set into application, then throughput can only be effectively improved by improving the data ingestion component of the algorithm. In a large-scale inference project, the constraints can be addressed in some cases by increasing the number of virtual components and GPU resources made available to the NVIDIA Clara scheduler. Therefore, increasing the number of CPUs and GPUs assigned can then be used by the NVIDIA Clara scheduler to efficiently keep the optimal number of simultaneous jobs running in the pipeline. Optimizing throughput for a large workflow through configuration and planning ultimately decreases the batch job time through parallel operations and thus allows processing of a large number of studies efficiently.

Best Practices - Data Considerations

In addition to an adequate data pipeline, effective model evaluation comprises careful consideration of cohort selection, ground truth annotations and statistical analysis. In the next two sections, we have put together a set of best practices that we have found useful in our existing work.

Accounting for the spectrum of disease severity

Using validation data without adequate representation risks resulting in performance that does not reflect the model's true performance if deployed in clinical settings. To determine whether the model is appropriately designed for its target population, a validation set that accurately captures the spectrum of disease severity is essential. All machine learning models that aim to detect findings are subject to the trade-off between correctly flagging positive cases (sensitivity) and avoiding flagging negative cases (specificity). This trade-off links in with the positive and negative predictive values, which are also dependent on how common positive cases are in the population. Suppose we have a model that detects pneumothorax on chest x-ray images. One of the key radiographic findings for a pneumothorax, which results from a punctured lung, is the absence of lung markings in the region where the lung should be. Naturally, a larger pneumothorax will have a greater region of absent lung markings and will likely be easier for the model to detect. If a model were to only be tested on large pneumothoraces then it may have a perceived elevated sensitivity compared with small pneumothoraces that it will likely encounter in the clinical environment.

This spectrum of disease severity also feeds its way back to model development, when it can impact parameters such as sampling strategy choice, batch size and operating point selection. It is not necessarily a simple choice as model training may benefit from overrepresentation or underrepresentation of particular cases. Furthermore, there may also be clinical considerations for when the disease severity reaches a threshold to impact management. We suggest looking at existing literature, reviewing clinical data and talking to stakeholders to consider the spectrum of disease severity.

Selecting diverse validation data

Having a diverse validation dataset that appropriately represents the target population and clinical environment is key for demonstrating model generalizability and reduces the risk of algorithmic bias. Dimensions that should be considered include different clinical sites, acquisition protocols, technical parameters, demographics, disease severities, geographical areas and anatomical differences in size or shape. Our experience suggests that ensuring good representation of scanner manufacturers is especially critical. Depending on the environment where the final model will be deployed, it is also important to actively select an appropriate balance between inpatient, outpatient and emergency department data. While having perfect coverage across all dimensions listed above is ideal, it is typically not feasible to account for all measures. We suggest deciding which parameters are most likely to affect the model's performance when used clinically and then controlling for them. We also highlight that, while having a diverse validation dataset is important for the model overall, it is also important to perform local validation and algorithm calibration on data at sites using the model given the variability that can occur.

Performing ground truthing at large scale

Ground truthing for healthcare AI models is the process of obtaining data labels that can then be compared to model outputs. For radiology models, the most common method for defining ground truth involves radiologists interpreting the DICOM images. It can be applied to both classification and localization. Classification interpretations are often derived by consensus with a “2+1” approach such that two radiologists initially interpret a case; if their interpretations are concordant then the interpretation is considered final and if their interpretations are discordant then a third radiologist provides an adjudicating interpretation. Localization interpretations vary in granularity with the spectrum including a point of interest (e.g., centroid), a bounding box or an intricately segmented region; they often require more consideration for how to achieve consensus especially as the intersection versus union of two radiologists’ regions (i.e., “and” versus “or” functions) can result in two quite different regions. Sometimes the ground truth interpretation may be defined on a different diagnostic test (e.g., a biopsy for a mass lesion) or on a patient’s clinical diagnosis including consideration of relevant coding systems such as International Classification of Disease (ICD). For imaging-based AI, there are other sources of information that may be helpful for large-scale ground truthing, such as radiology reports leveraging natural language processing. Models that use localization interpretations typically require a manual ground truthing process, which may constrain the size of the validation data sets. When available and appropriate for the modeling context, electronic health data may provide a valuable source of ground truth for large populations. An example of consideration of broader clinical situation is the CO-RISK model for COVID-19 patients (see *Use Cases*), which uses both chest x-ray images and structured data, including demographics and hematological lab values, to predict patient needs for oxygen devices.

There are ongoing efforts to make radiology reports and the underlying data more structured. The American College of Radiology provides the ACR Assist for Computer Assisted Reporting and Decision Support, which encodes guidelines from medical societies on best practices for reporting and follow-up recommendations for specific conditions [10, 11]. This application can be utilized to derive data for ground truth interpretations as part of model validation (or training sets for model development). It can also ingest output data from AI models in a structured manner. These efforts support large-scale ground truthing and AI results feedback based on routine clinical interpretations at the point of care and are currently in use at Mass General Brigham.

Counteracting data drift

As the context of a deployed model changes, the characteristics of the input data may evolve to a point where model performance decreases, which is commonly known as *data drift*. Potential strategies for avoiding this decline include reducing the time from training to deployment and validating the model on a population that accurately reflects the intended clinical environment. Upon deployment, it is crucial to continue monitoring the model closely including comparing predictions with the ground truth. The underlying source for any reductions in performance can potentially be addressed in close collaboration with clinicians, who can provide insight into potential changes such as treatments, protocols, scanners, or case trends. We suggest automating the analysis of model predictions to the greatest extent possible and striving to find methods of receiving feedback from models deployed clinically without adding to the workload of physicians.

Best Practices - Statistical Analysis Considerations

The Transparent Reporting of a multivariable prediction model for Individual Prognosis or Diagnosis (TRIPOD) guidelines provide statistical recommendations for the reporting of studies developing, validating, or updating a prediction model [12]. The TRIPOD recommendations can be extended to the large-scale validation setting, with a few additional considerations and challenges.

Common measures of the predictive performance of a model are discrimination and calibration. They refer to the ability of a model's predicted risks to separate patients that do and do not have the disease or event of interest (discrimination), and the agreement between the predicted and observed risks (calibration) [13]. Discrimination is usually measured by the C-statistic (equivalent to the Area Under the Receiver Operating Characteristic Curve – AUC – in the logistic regression case), and model calibration is quantified by measures such as the calibration intercept, calibration slope, and the expected/observed statistic. These performance measures should be estimated for small or large-scale model validation.

Since large-scale or "big" data are becoming increasingly available to validate published prognostic models, Riley et. al. suggest opportunities to improve validation using big data through the following set of examples [13]:

1. Examining consistency in a model's predictive performance across multiple studies
2. Examining consistency in performance across multiple practices
3. Examining performance in clinically relevant subgroups
4. Examining sources of heterogeneity in model performance
5. Examining model recalibration strategies (model updating)

These examples outline the many ways large-scale data can make the validation process more rigorous compared to small-scale validation.

Additionally, large-scale data may require additional data quality checks including missing data, non-standardized definitions of model variables and outcomes, incomplete follow-up and event dates, and lack of recording of potentially important or novel predictors [14]. In particular, missing data can lead to multifaceted obstacles during model validation. The type of missingness needs to be investigated and understood for every model variable. If a variable is missing at random then advanced imputation techniques can be implemented to fill in the missing values to obtain a risk score [14, 15]. Missing data can also be considered in the model development phase, by fitting a unique prediction model for every possible missing data pattern that is fit using only data from that pattern [16].

A set of extensions to the existing TRIPOD guidelines has been proposed specifically for validation studies employing large datasets. It includes providing a description of how data sources were identified and, for studies using EMR data, an overview of the data gathering process including how any data joins were conducted. Data clustering due to distinct treatment protocols or involvement of multiple hospital sites should be clearly documented and differences in image acquisition protocols, case types or classification practices should be outlined. If quantitative predictors are used, it is appropriate to display cluster-wise summary statistics for the baseline characteristics and describe methods used to handle any discrepancies between clusters in how predictors were defined or captured. Results should be presented for the full validation dataset, each cluster and any relevant clinical or demographic subgroups. Techniques used for any subsequent recalibration of the model should be reported together with any positive (or negative) changes in performance. [13]

Large scale validation provides the opportunity to better understand a model's predictive performance (discrimination and calibration) across a variety of datasets, clinical settings, and patient demographic groups. It also offers a setting to evaluate the deficiencies that exist within a model and if additional methodological strategies (prediction submodels, recalibration) need to be implemented to make the prognostic and diagnostic models more generalizable.

Use Cases

Case Study: Breast Cancer Risk Prediction

Accurate breast risk cancer prediction enables the use of targeted screening strategies that improve the chance of early detection. A team of researchers at Massachusetts Institute of Technology, Harvard Medical School and Massachusetts General Hospital (MGH) developed a mammography-based deep learning model for breast cancer risk prediction, which was externally validated and published in *Science Translational Medicine* and *Journal of Clinical Oncology* in 2021 [17, 18]. Using the four standard views of a mammography study (left/right + craniocaudal/mediolateral oblique), the model produced a prediction of a woman's risk of breast cancer in the next 1, 2 and 5 years.

Following the external validation, an inference pipeline was developed to enable large-scale processing of this AI risk prediction model on consecutive screening mammograms obtained in the MGH system from 2009 - 2021, as well as on consecutive screening mammograms from African American and Hispanic women in BWH imaging clinics between 2005 - 2014. This resulted in over 450,000 AI risk scores generated from MGH-system screening mammograms, and nearly 60,000 AI risk scores generated from BWH-system screening mammograms. Developing, customizing, and refining the infrastructure for this inference pipeline took place over several months and involved close collaboration between the research lab (MGH Breast Imaging Research Center) and the core technology center hosting and managing the pipeline (Mass General Brigham Enterprise Medical Imaging). The final, implemented version of the inference pipeline utilized Laurel Bridge Compass for the DICOM routing of exams, which filtered out all studies missing any of the four required views. Five NVIDIA Clara instances on separate servers were used for model processing, and the resulting AI-based risk scores were stored in a results database. Challenges in processing the large datasets included handling image series descriptions, which had evolved over time, and server version inconsistencies. An exam-code mapping scheme was developed within Compass to address the series description discrepancies and ensure consistent series descriptions across historical variations. Jobs were tailored to match the capacity of each of the five servers through the addition of explicit 1-minute intervals between studies and continuous disk space cleaning. Once optimized, the final system was able to process and return AI risk scores at a consistent rate of approximately 5,000 studies per day, with peak performance of up to 6,000 exams daily. We expect that by altering the model and system architecture to take advantage of GPU-based inferencing and data processing unit (DPU)-based data transfer, removing overhead required for model loading, for example using NVIDIA Triton, we would be able to maintain this throughput using fewer servers and Clara instances or, alternatively, higher throughput with the same number of instances using GPUs.

Case Study: COVID-19 Pulmonary X-Ray Severity (PXS) Model

The purpose of this study was to evaluate a model for quantification of pulmonary disease severity in COVID-19 patients from anterior-posterior chest X-rays. The development and initial evaluation of this model had already been described in a peer-reviewed scientific manuscript in the journal Radiology: Artificial Intelligence [19]. The model was based on a Siamese convolutional neural network model and utilized approximately 160,000 chest x-rays from the public CheXpert dataset for weakly-supervised pre-training followed by fine-tuning on 314 x-rays from COVID-19 patients from our institution with severity grades assigned by radiologists. The initial evaluation in the manuscript on data from a test set of images from our institution (154 images) and another institution (113 images) demonstrated both agreement of the model's severity score with those of radiologists and the utility of the model's score in predicting subsequent intubation or death within 3 days of hospital admission.

After the model development and initial evaluation were completed, the GPU-based model was deployed at Mass General Brigham in order to further validate its performance in a clinical setting. The Compass DICOM routing software was configured to send chest radiograph studies to the model hosted in NVIDIA Clara, and specifically select the anterior-posterior (AP) image for analysis. The generated PXS score severity score was stored in the AI results database for analysis. Over a time period of 567 days beginning in July 2020, the model was successfully run on 159,123 imaging studies, which averages 280 studies each day. On the busiest day during this period 1,116 imaging studies were run through the model.

The results demonstrated that the PXS model was effective at monitoring COVID surges and reducing inter-reader variability between radiologists assessing disease severity. Furthermore, the numerical severity results gathered on all chest x-rays at the hospital provided a valuable resource for researchers studying the new disease as the pandemic unfolded, especially when combined with other data sources. For example, using this data gathered during this time, researchers have demonstrated that right ventricular strain is not associated with the severity of respiratory illness, but that the risk of acute neuroimaging findings is associated with the severity of respiratory illness [20, 21].

Case Study: CO-RISK model

Effective triaging of COVID-19 patients relies on the identification of possible worse prognosis (needing advanced oxygen therapy, mechanical ventilator, or have a higher risk of death) at the patient's first presence to the Emergency Department. At MGH, a multidisciplinary team from the Department of Radiology and Emergency Department developed a deep learning-based predictive model ("CO-RISK") for severe COVID-19 outcome based on a large-scale analysis of 11,060 consecutive COVID patients who had been admitted to one of five Emergency Departments in the Greater Boston area [22]. The model takes the full spectrum of data available at initial presentation in the Emergency Department as input, including EMR data such as demographics, vital signs and lab results, as well as chest x-ray imaging studies, and generates the CO-RISK score for stratifying short-term (24/72h) needs of interventions (oxygen therapy required). CO-RISK score achieved AUC for predicting severe outcomes in 24 hours of 0.95 and in 72 hours of 0.92. Compared with physicians' decisions, CO-RISK score demonstrated superior performance in making decisions about ICU/floor admissions [22].

After the CO-RISK model was developed and validated on an external validation dataset, a lightweight version of the model, using the 5 most critical variables, was deployed at the MGH Emergency Department and tested prospectively in real time. By acquiring EMR data through Epic FHIR interconnect (DSTU2) and imaging data through Compass, the model could process and make the corresponding prediction for around 1,000 cases per day. Success in the CO-RISK model deployment and connection to institutional data resources motivated further international multi-institutional study (the “EXAM” study) for COVID risk prediction in collaboration with NVIDIA [23].

Conclusion

The exciting integration of AI into clinical workflows calls for rigorous model validation practices. While nearly all FDA-cleared AI/ML algorithms used less than 1000 studies for validation, the proposed framework allows for using validation datasets several orders of magnitude larger involving handling increased processing efforts, large-scale series selection and ground truthing. Diverse datasets comprising a range of demographic, technical and clinical characteristics are critical for confirming that a model operates successfully in real hospital settings. By employing large-scale batch inference early in the development process and ensuring that a model is validated on data representative of the patient population where it will be deployed, it is more likely that clinical performance will reproduce performance results seen during model development. Large-scale validation can also help to promote health equity.

Future directions

Validation of AI models is constantly evolving and regulatory bodies around the world are in the process of developing corresponding legislation. The FDA proposed a regulatory framework for AI and machine learning-based software as a medical device in April 2019 and subsequently released an Action Plan in January 2021 [24]. We anticipate updated legislation in the coming years and simultaneously new technical advances requiring further regulatory considerations. While all AI models currently approved by the FDA are 'locked', in the sense that they require additional clearance after updates, going forward we expect a movement towards autonomous AI devices continuously learning from live clinical data. This learning will likely help counter data drift, but also poses multiple potential challenges, such as the ones related to risk mitigation and model monitoring. A possible scenario is first introducing unlocked AI models on the operational side of healthcare before pursuing adaptive modeling directly tied to health outcomes [25].

We hope to see technical progress pertaining to post-market surveillance, as the current practice is mostly limited to manual checks on a product-by-product basis. There is a clinical need to develop a framework for systematically assessing model performance across diverse parameters, such as demographics or study protocols. It includes identifying failure modes, leveraging the knowledge of model misclassifications for retraining, and evaluating how any changes affect the model performance. We look forward to seeing a transition from the current static evaluation process to a more fluid model validation approach, while ensuring quality and patient safety.

The large-scale model validation process presented in this white paper uses a scalable end-to-end AI platform for large-scale inference. Potential future directions include developing pipelines that use GPU and multi-GPU operations, instead of CPU-only, for even faster throughput and execution. Image pre-processing and I/O can similarly be GPU-optimized with libraries for popular data science tools, such as Numpy and Pandas, which can be accelerated with NVIDIA cuDF and RAPIDS libraries that enable the execution of end-to-end pipelines entirely on GPUs [26]. Further process optimization for large-scale validation projects can be achieved using new AI deployment platforms, such as MONAI Deploy [27], which is open-source software that evolves from Clara. It takes advantage of almost any underlying compute resource and uses AI code to process the workloads accordingly.

References

1. AI Central. American College of Radiology Data Science Institute website. https://aicentral.acrdsi.org/. Accessed January 4, 2022.
2. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. Imagenet: A large-scale hierarchical image database. 2009 IEEE conference on computer vision and pattern recognition, pp. 248-255. Ieee. Accessed January 4, 2022.
3. Kim DW, Jang HY, Ko Y, Son JH, Kim PH, Kim SO, Lim JS, Park SH. Inconsistency in the use of the term "validation" in studies reporting the performance of deep learning algorithms in providing diagnosis from medical imaging. Plos one. 2020 Sep 11;15(9):e0238908.
4. Generating Evidence for Artificial Intelligence-Based Medical Devices: A Framework for Training, Validation and Evaluation. World Health Organization website. https://www.who.int/publications/i/item/9789240038462. Published 17 November 2021. Accessed January 20, 2022.
5. Ebrahimian S, Kalra MK, Agarwal S, Bizzo BC, Elkholy M, Wald C, Allen B, Dreyer KJ. FDA-regulated AI algorithms: Trends, strengths, and gaps of validation studies. Academic radiology. 2022 Apr 1;29(4):559-66.
6. Voter AF, Larson ME, Garrett JW, Yu JP. Diagnostic accuracy and failure mode analysis of a deep learning algorithm for the detection of cervical spine fractures. American Journal of Neuroradiology. 2021 Aug 1;42(8):1550-6
7. Gallas B, Segui J. Evaluating Artificial Intelligence Devices at the FDA and Related Collaborations and Invitiatives. 2019 Imaging Informatics Summit. Accessed at https://ncihub.org/groups/eedapstudies/wiki/DeviceAdviceAIMLImaging/File:J.A.Segui.ACR.Informatics.2019.Slides.FINAL.pdf on Jan 10, 2022
8. Gauriau R, Bridge C, Chen L, Kitamura F, Tenenholtz NA, Kirsch JE, Andriole KP, Michalski MH, Bizzo BC. Using DICOM Metadata for Radiological Image Series Categorization: a Feasibility Study on Large Clinical Brain MRI Datasets. J Digit Imaging. 2020 Jun;33(3):747-762. doi: 10.1007/s10278-019-00308-x. PMID: 31950302; PMCID: PMC7256138.
9. Krishnan K, Wyman J, 2021. Clara Python Client. Github. Published on June 28,2021. Accessed at https://github.com/NVIDIA/clara-platform-python-client on February 10, 2022.
10. Alkasab TK, Bizzo BC, Berland LL, Nair S, Pandharipande PV, Harvey HB. Creation of an open framework for point-of-care computer-assisted reporting and decision support tools for radiologists. Journal of the American College of Radiology. 2017 Sep 1;14(9):1184-9.
11. ACR Assist. American College of Radiology. Accessed at https://assist.acr.org/ on January 20, 2022.
12. Moons KG, Altman DG, Reitsma JB, Ioannidis JP, Macaskill P, Steyerberg EW, Vickers AJ, Ransohoff DF, Collins GS. Transparent Reporting of a multivariable prediction model for Individual Prognosis or Diagnosis (TRIPOD): explanation and elaboration. Annals of internal medicine. 2015 Jan 6;162(1):W1-73.
13. Riley RD, Ensor J, Snell KI, Debray TP, Altman DG, Moons KG, Collins GS. External validation of clinical prediction models using big datasets from e-health records or IPD meta-analysis: opportunities and challenges. bmj. 2016 Jun 22;353.
14. Jolani S, Debray TP, Koffijberg H, van Buuren S, Moons KG. Imputation of systematically missing predictors in an individual participant data meta-analysis: a generalized approach using MICE. Stat Med 2015;34:1841-63. doi:10.1002/sim.6451.
15. Resche-Rigon M, White IR, Bartlett JW, Peters SA, Thompson SG. PROG-IMT Study Group. Multiple imputation for handling systematically missing confounders in meta-analysis of individual participant data. Stat Med 2013;32:4890-905. doi:10.1002/sim.5894
16. Fletcher Mercaldo S, Blume JD. Missing data and prediction: the pattern submodel. Biostatistics. 2020 Apr;21(2):236-52.
17. Yala A, Mikhael PG, Strand F, Lin G, Smith K, Wan YL, Lamb L, Hughes K, Lehman C, Barzilay R. Toward robust mammography-based models for breast cancer risk. Science Translational Medicine. 2021 Jan 27;13(578):eaba4373.

18. Yala A, Mikhael PG, Strand F, Lin G, Satuluru S, Kim T, Banerjee I, Gichoya J, Trivedi H, Lehman CD, Hughes K. Multi-institutional validation of a mammography-based breast cancer risk model. Journal of Clinical Oncology. 2021 Nov:JCO-21
19. Li MD, Arun NT, Gidwani M, Chang K, Deng F, Little BP, Mendoza DP, Lang M, Lee SI, O'Shea A, Parakh A. Automated assessment and tracking of COVID-19 pulmonary disease severity on chest radiographs using convolutional siamese neural networks. Radiology: Artificial Intelligence. 2020 Jul 22;2(4):e200079.
20. Gibson LE, Fenza RD, Lang M, Capriles MI, Li MD, Kalpathy-Cramer J, Little BP, Arora P, Mueller AL, Ichinose F, Bittner EA. Right ventricular strain is common in intubated COVID-19 patients and does not reflect severity of respiratory illness. Journal of intensive care medicine. 2021 Aug;36(8):900-9.
21. Lang M, Li MD, Jiang KZ, Yoon BC, Mendoza DP, Flores EJ, Rincon SP, Mehan WA, Conklin J, Huang SY, Lang AL. Severity of chest imaging is correlated with risk of acute neuroimaging findings among patients with COVID-19. American Journal of Neuroradiology. 2021 May 1;42(5):831-7.
22. Buch V, Zhong A, Li X, Rockenbach MA, Wu D, Ren H, Guan J, Liteplo A, Dutta S, Dayan I, Li Q. Development and validation of a deep learning model for prediction of severe outcomes in suspected COVID-19 Infection. arXiv preprint arXiv:2103.11269. 2021 Mar 21.
23. Dayan I, Roth HR, Zhong A, Harouni A, Gentili A, Abidin AZ, Liu A, Costa AB, Wood BJ, Tsai CS, Wang CH. Federated learning for predicting clinical outcomes in patients with COVID-19. Nature medicine. 2021 Oct;27(10):1735-43.
24. Proposed regulatory framework for modifications to artificial intelligence/machine learning (AI/ML)-based software as a medical device (SaMD). Food and Drug Administration. Published in 2019. Updated on September 22, 2021. Accessed on January 4, 2022.
25. Pianykh OS, Langs G, Dewey M, Enzmann DR, Herold CJ, Schoenberg SO, Brink JA. Continuous learning AI in radiology: implementation principles and early applications. Radiology. 2020 Oct;297(1):6-14
26. RAPIDS - Open GPU Data Science. Github. Accessed at https://github.com/rapidsai on January 10, 2022.
27. MONAI Deploy. Github. Accessed at https://github.com/project-monai/monai-deploy-app-sdk on August 5, 2022.

Appendix

Example 1: YAML file for pipeline configuration

```
  api-version: 0.4.0
  orchestrator: Argo
  name: example-pipeline
  operators:
  - name: preprocessing
    description: A container to perform image preprocessing
    container:
      image: example-docker-registry/preprocessing
      tag: v0.1
    requests:
      cpu: 2
      memory: 2GB
    input:
    - path: /input/
    output:
    - path: /output/
  - name: inference-container
    description: A container that performs inference
    container:
      image: example-docker-registry/inference-container
      tag: v0.1
    requests:
      cpu: 8
      memory: 8GB
      gpu: 1
    input:
    - from: preprocessing
      path: /input/
    output:
    - path: /output/
  - name: send-results
    description: Reads results, creates JSON results, and sends it to the
database.
    container:
      image: example-docker-registry/inference-results-sender
      tag: v0.1
      command: ['python', 'example.py', '--url', 'http://my-output-
example.com/inference-results']
    requests:
      cpu: 2
      memory: 2GB
    input:
    - from: inference-container
      path: /input
```

Example 2: Use of NVIDIA Clara command-line tools to complete pipeline configuration

```
    user@clara-server:~$ clara create pipeline -p /path/to/pipeline-
example.yaml
    {output}
    user@clara-server:~$ clara dicom create aetitle -a EXAMPLE_INF -o
pipeline-example={pipelineid}
    user@clara-server:~$ clara dicom create source -a MYSTORESCU -i
{ip_address}
```

Now that the pipeline has been configured, NVIDIA Clara will automate the process when a study is sent to the AE title, “EXAMPLE_INF”. The data is then passed to each of the operators in the pipeline. Activity on this pipeline can be monitored and troubleshot using command line tools.

Example 3: Activity monitoring using the command line tools.

```
kubectl get po
kubectl logs {containername}
clara list jobs
clara describe job -j {job id}
```

Finally, the example below demonstrates how to create a “Running Jobs” gauge.

Example 4: Creation of a Running Jobs gauge

```
from nvidia_clara.jobs_client import JobsClient
import nvidia_clara.job_types as job_types
import plotly.graph_objects as go

clara_ip_address = "127.0.0.1"
clara_port =  "30031"
running_job_filter =
job_types.JobFilter(has_job_state=[job_types.JobState.Running])
run_job_list = jobs_client.list_jobs(job_filter=running_job_filter)
fig = go.Figure(go.Indicator(mode="gauge+number", value= len(run_job_list),
domain = {'x': [0, 1], 'y': [0, 1]},title = {'text': "Running Jobs"},
gauge={'axis':{'range':[0,20]}}))

fig.show()
```