Real-time Multi-Modality Clinical Decision Support Platform: An Overview of Incorporating Deep Learning within Multi-Modality Fusion Framework in HealthCare

A. Kia*, P. Timsina**

*Mount Sinai Health System, NY, NY 10029, arash.kia@mssm.edu
**Mount Sinai Health System, NY, NY 10029, prem.timsina@mssm.edu

ABSTRACT

In this paper, we will present the blueprint for a multi-modality decision support platform. Modern health care systems are generating clinical data at an ever-increasing rate and in a wide range of formats including structured, unstructured, image, and high-frequency waveform data. The volume of data is also growing at an exponential rate. Analyzing these silos of real-time clinical data and providing timely and personalized clinical interventions are challenging tasks. To address this problem, we developed a platform for incorporating deep learning within a multi-modality fusion framework, whereas any updates to the electronic health record (EHR) platform stream to a data science engine that generates real-time predictive notifications. We deployed the platform in a large healthcare system across multiple hospitals in New York City and conducted extensive evaluations. Here, we describe the infrastructure, and practical lessons learned from deploying real-time and batch ML applications across multiple hospitals.

Keywords: Machine Learning Platform, Multi-modality, Deep learning, Healthcare

1 INTRODUCTION

Real-time clinical decision support (CDS) tools using stream-data processing allow clinicians to acquire deeper knowledge about their patients in a timely manner [1]. This has the potential to enable earlier clinical interventions that improve quality of care, enhance patient safety, and minimize hospital morbidity and mortality [2]. The clinical data required to provide such decision support are varied, large, and mainly in an unstructured format. The data sources are multi-modal—data coming from electronic health record (EHR) platforms, genome sequences, Medical imaging platform, and patient-generated home healthcare data mainly coming from wearable devices. Frequently, a patient's real-time data must be combined with historical patient information to generate a complete picture of the patient's current clinical situation. To use data from multiple sources for machine learning (ML), they must be transformed into a standardized data representation called ML features. Furthermore, once ML features have been created, a wide range of ML techniques must be applied to generate ML predictions. These ML steps—from data ingestion to modeling and putting ML software into production — are an iterative process that requires multiple prototyping. Consequently, a standardized ML pipeline that serves as a platform for data ingestion is warranted for building ML software before finally operationalizing it.

In this paper, we describe the framework for developing an ML pipeline that includes multi-modal data, and computationally expensive machine learning processes. To build the framework, we used big-data open-source technology stacks such as Mirth Connect [3], Apache Kafka [4], MongoDB [5], and Apache Spark [6]. Additionally, we deployed the platform as a hybrid infrastructure that includes on-premise and cloud (Microsoft azure).

2 Objective

The overall aim of this study was to provide a blueprint to develop a Multi-Modality Clinical Decision Support Platform and deploy it in a hibrid computational infrastructure. Thus, the objectives were as follows:

- To present the end-to-end architecture for an ML pipeline using readily available, industry-standard, and cost-effective technology stacks.
- To provide the architecture for including multi-modal data and various machine learning techniques to process such data

3 SYSTEM REQUIREMENTS

The requirements for the proposed architecture were synthesized and generalized from a supporting literature review [7] [8] as well as the authors' experience in developing and deploying ML pipelines. Inputs from multiple discussion sessions with a data scientist, data engineer, system engineer, and clinical experts were used to validate the requirements. In effect, the proposed architecture should support the following requirements:

Scalability*:** The system should be highly scalable and able to dynamically scale up and down. ***Security: The overall system should be a HIPAA compliant infrastructure and allow secure data transfer and storage. ***Heterogeneous Data Ingestion, Storage, and Consumption:*** *The* ML pipeline should be able to ingest, store, and consume data in multiple formats, such as structured, free-text, and Medical image data. ***Reusability*:** The core components and functionalities

of the pipeline should be reusable across multiple projects. ***Automation*:** The pipeline should minimize manual tasks and support continuous deployment in both on-premise and cloud to minimize the network bandwidth and latency for data integration and real-time engine. We found that it is more cost efficient to run the relatime applications and data services on-premise.

The major components that are deployed in the cloud are Continous Integration/Continous Deployment (CI/CD) pipelines using Azure dev-ops , batch machine learning engine, and high performance computation tasks such as image processing and deep learning piplines.

Requirements	Components	Supported Features
Scalability	Data lake	The MongoDB database has a sharded cluster, which supports horizontal scaling.
	Computation environments	Microsoft Azure Spark Cluster is configured to automatically scale up or down based on computation load. Batch Jobs create new Spark Clusters and upon completion of jobs kill the cluster; this allows efficient scaling.
	Messaging	Apache Kafka cluster supports horizontal scaling by adding machines based on incoming message rate.
Heterogeneous Data Ingestion, Storage, and Consumption	DS engine	The DS engine is capable of ingestion and consumption of streaming messages (HL7, JSON, and Text), and bulk pull from multiple database systems (e.g., MYSQL, MongoDB, Hive, and Oracle).
Reusability	Codes	The MOUNT SINAI Data Science Toolkit allows optimal reusability of codes. Functionalities such as normalization, feature engineering, and feature transformations are written, validated, packaged, and added to this toolkit. All the use cases utilized this toolkit to create the ML engine.
Automation	Software deployment	We containerized the project into a Docker container, which is able to run on any operating system (e.g., Windows, macOS, Linux, and Cloud) and in any environment (i.e., cloud vs. local deployment).
	Monitoring, troubleshooting, and debugging	We used Nagios, Splunk, and a custom Python dashboard to continuously monitor the infrastructures and applications. Upon encountering errors, the system attempts to fix the error itself (such as through rebooting).
Modularization	Codes	Functionalities are organized as packages in the DS toolkit.
Reproducibility		We used the customized Lambda architecture so that processed data from both batch and stream paths produce the same results.

Table 1. Requirements and Functionalities

4 RESULT

The ML platfom with a MongoDB data lake, Spark clusters, Batch ETL pipelines, and ML applications went live in August 2017 feeding the stream of data from one hospital of Mount Sinai Health Systems(MSHS). We went through multiple iterations; and the hybrid infra with on-premise and cloud was deployed in Aug 2021. Currently, the platform is deployed across 6 hospitals within new york city. The current data volume is about 10TB which includes structured EHR data, imaging data, and clinical free-text notes. Additionally, the data integration engine is receiving 60 million HL7 messages and 50,000 images per hour. The platform hosts 30 machine learning apllications deployed across 6 hospitals.

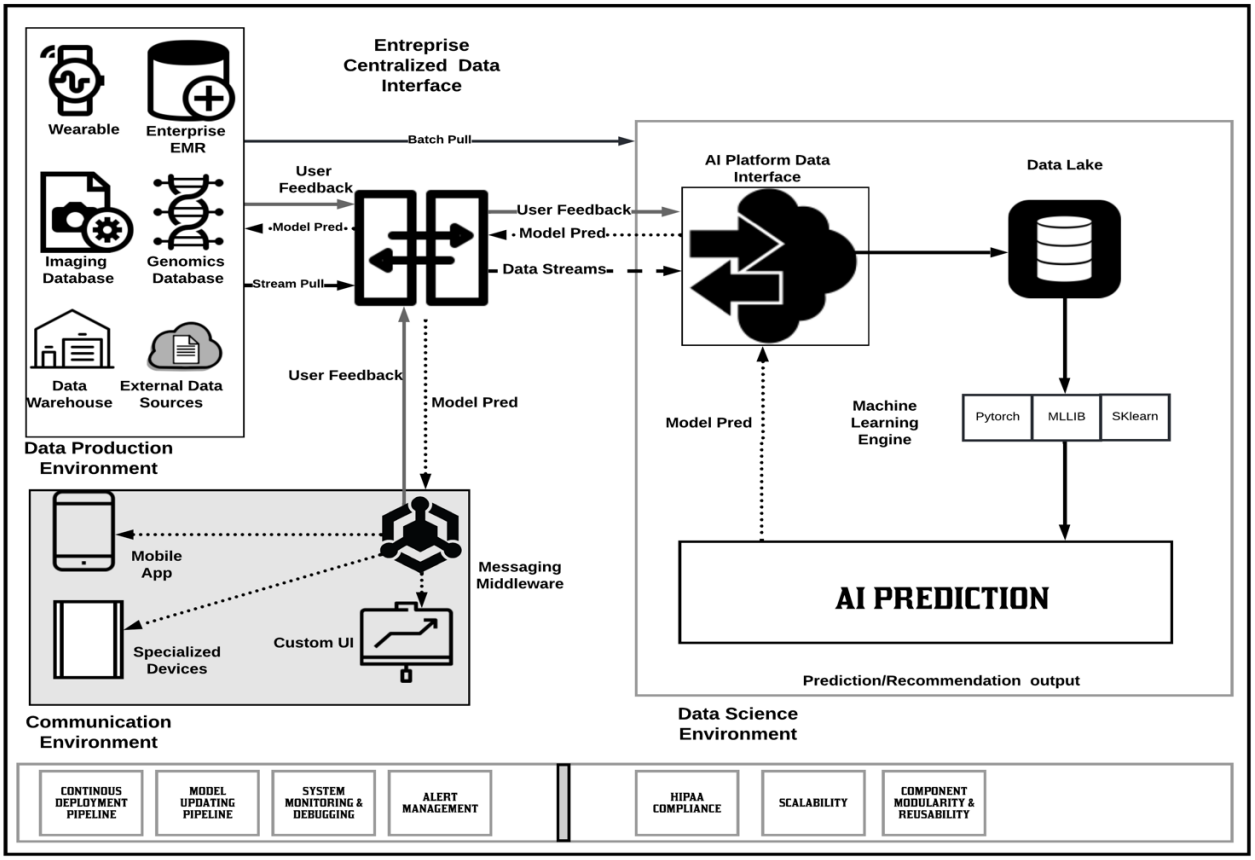

Figure 1. Multi-Modality ML Platform Architecture

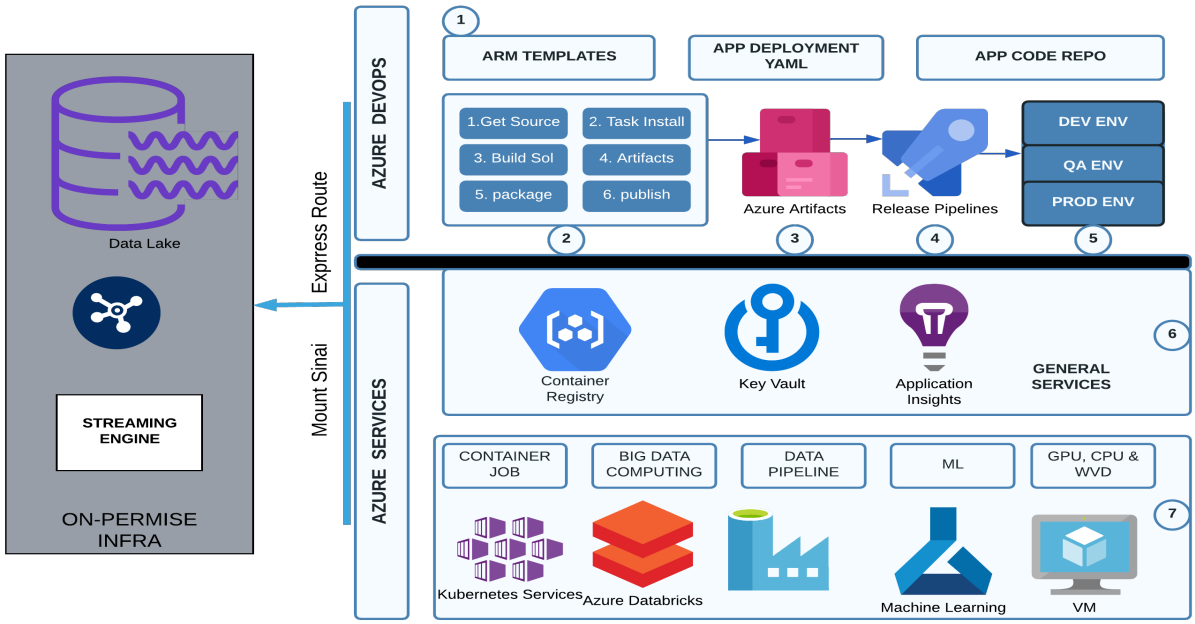

Figure 2. Hybrid Architecture

5 DISCUSSION

In this paper, we illustrated the architecture for a real-time ML Platform for healthcare systems. In addition, we illustrated the implementation details and perspective results of two ML engines. We faced three major challenges in the creation of the streaming ML pipeline: (1) **Multiple data sources and the lack of a unified data dictionary across EHR systems:** To address this problem, we created a customized data normalization software package that can take in a wide variety of data formats and convert them into standardized data representations. For example, our normalization package has methods to convert different medication coding formats into the international classification coding system (e.g., RxNorm). (2) **Data quality:** Most data were highly sparse and qualitative in nature, which required considerable time for data preparation. To overcome data quality challenges, we built a Data Quality Monitoring System that monitors the veracity of incoming data and notifies the engineering team of problems in data quality. (3) **Optimal processes not yet determined:** This real-time health care ML pipeline is still in the early phase of its development, and mature blueprints to create such a pipeline do not exist. We have gone through multiple iterations and conducted extensive prototyping to create a reliable, flexible, and scalable ML pipeline.

The lesson we learned from

6 CONCLUSION

Real-time CDS systems require a robust infrastructure that is capable of consuming a variety of historical and streaming data, conducting feature engineering and transformation, and building ML softwares before finally operationalizing it. In this paper, we described a blueprint for such a system that is suitable for mid and large-scale healthcare systems. Unlike earlier relevant studies that have tackled only parts of the problem, such as model building and data ingestion, we proposed end-to-end design guidance for researchers and practitioners to develop a real-time ML infrastructure for mid and large-scale healthcare systems. To develop such an infrastructure, we majorly employed open-source technology that is freely and readily available to anyone.

7 REFERENCES

[1] H. Lin, H. Wu, C. Chang, T. Li, W. Liang, and J. W. Wang, "Development of a real-time clinical decision support system upon the web mvc-based architecture for prostate cancer treatment," *BMC Med. Inform. Decis. Mak.*, vol. 11, no. 1, p. 16, 2011.

[2] D. Jones, I. Mitchell, K. Hillman, and D. Story, "Defining clinical deterioration," *Resuscitation*, vol. 84, no. 8, pp. 1029–1034, 2013.

[3] NextGen, "Mirth Connect," 2022. [Online]. Available: https://www.nextgen.com/products-and-services/integration-engine. [Accessed: 14-Apr-2022].

[4] Apache Softwarre Foundations, "Apache Kafka," 2022 .

[5] MongoDB, "MongoDB." .

[6] Apache Spark, "Apache Spark." [Online]. Available: http://spark.apache.org. [Accessed: 14-Apr-2022].

[7] IBM, "A reference architecture for high performance analytics in healthcare and life science," no. March, pp. 1–17, 2017.

[8] M. A. Levin *et al.*, "iGAS: A framework for using electronic intraoperative medical records for genomic discovery," *J. Biomed. Inform.*, vol. 67, pp. 80–89, 2017.

[9] Infor, "Cloverleaf Integration Suite," *infor*, 2018.

[10] Microsoft Azure, "Microsoft Azure", [Online]. Available: https://azure.microsoft.com/en-us/. [Accessed: 14-Apr-2022].