P-DASION

Aimee Co, James Duong, Mildred Morales-Paredes
Shadab Sharif, Wendi Zheng, Aleicia Zhu

Enhancing OpenEMR with advanced AI/ML integration

PROBLEM STATEMENT

The current Electronic Medical Records (EMR) systems are facing increasing demands for improved efficiency and effectiveness. EMR systems are vital tools in modern healthcare, serving as centralized platforms for storing, managing, and sharing patient information. They enhance communication between healthcare providers, streamline workflows, reduce errors, and improve patient safety by ensuring accurate and up-to-date medical information is always available. OpenEMR, a widely-used and open-source EMR system, is aging and in need of modernization to meet the evolving needs of healthcare providers.

OBJECTIVE

Integrate advanced AI/ML algorithms into OpenEMR to enable real-time data analysis, helping healthcare providers make informed decisions with minimal delay.

- CODEBASE AND DATA FAMILIARIZATION
- LITERATURE REVIEW
- MODEL DEPLOYMENT
- CONNECT FRONT-BACK

DATA EXPLORATION

MIMIC-III (Medical Information Mart for Intensive Care) is a large, publicly available database containing de-identified health records of over 40,000 critical care patients. It includes a rich collection of structured and unstructured data, such as:

- Clinical Notes: Free-text records from nurses, physicians, and radiology, among others.
- ICD Codes: Standardized diagnostic and procedural codes assigned during patient care.
- Temporal Data: Timestamped records that capture patient interactions throughout their hospital stay.

CONTINUOUS ICD PREDICTION

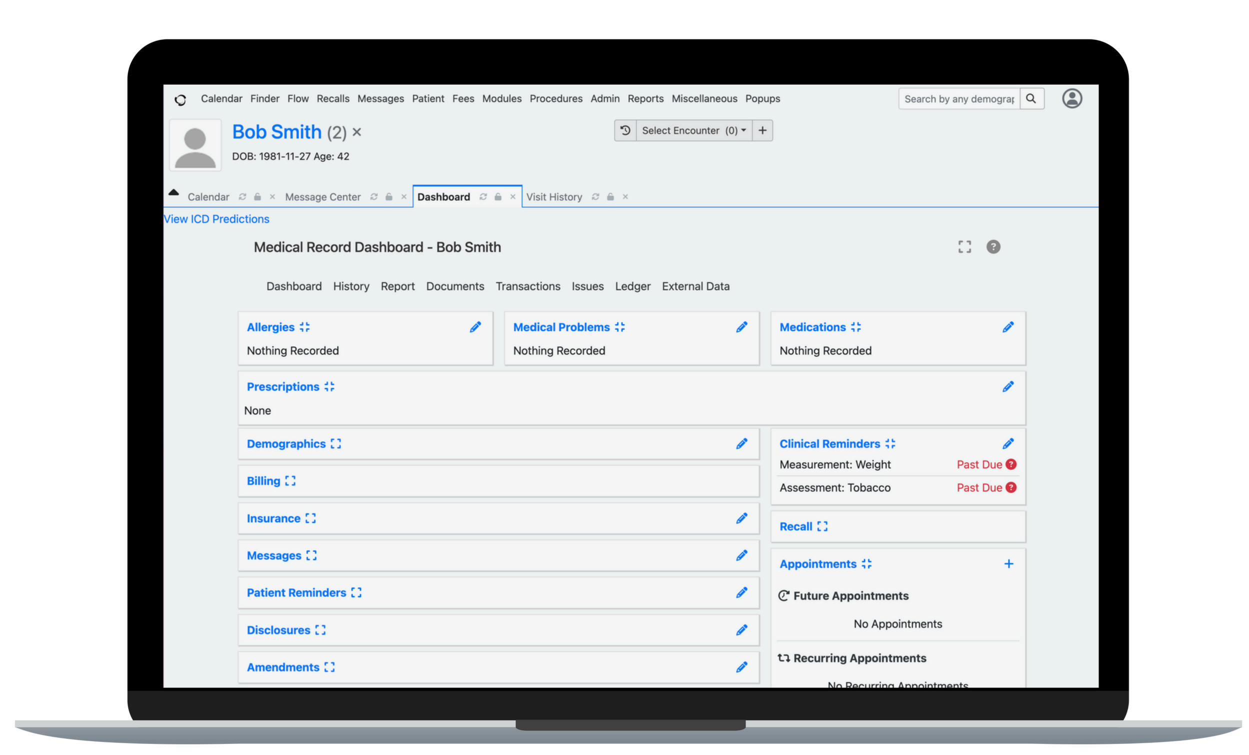
This is a novel approach for predicting International Classification of Diseases (ICD) codes at various stages during a hospital stay, using early clinical notes from electronic health records (EHR). The model, Label-Attentive Hierarchical Sequence Transformer (LAHST), is designed to efficiently predict ICD codes using only the information available up to a specific time point, without relying on post-discharge summaries.

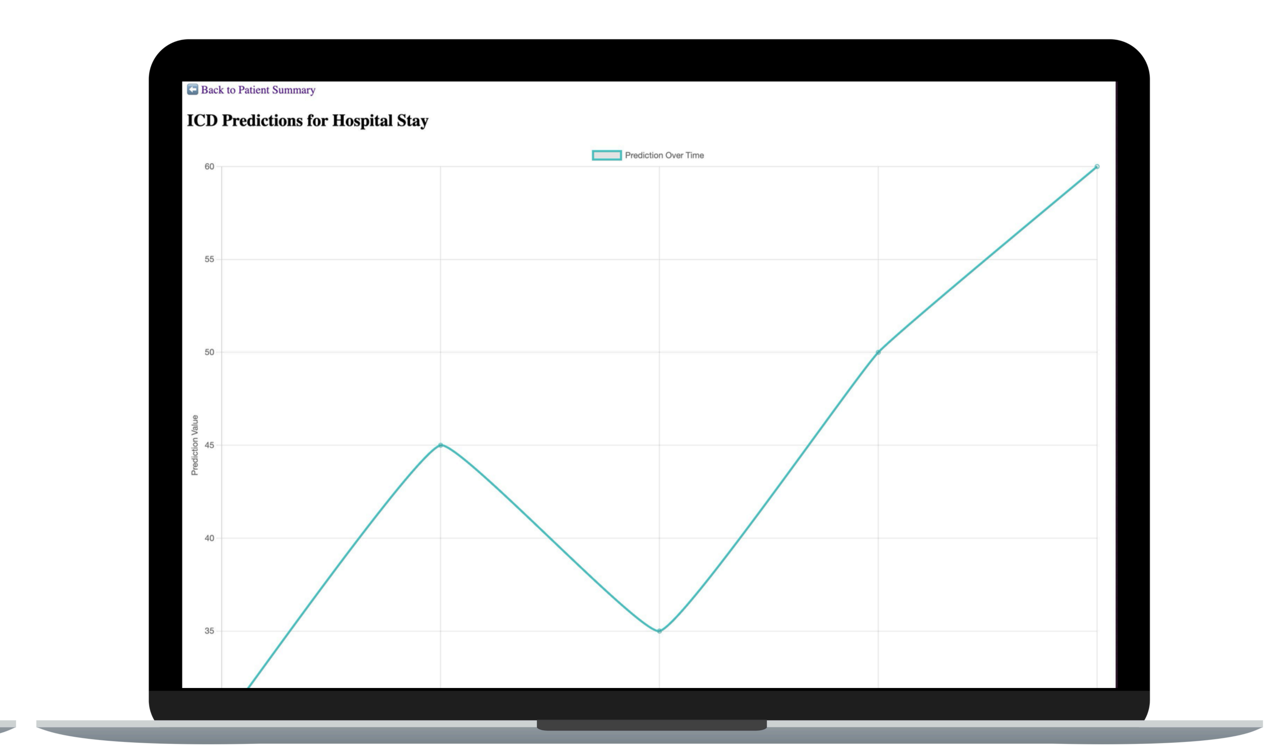
- The model enables predictions as early as two days after admission, achieving a strong predictive accuracy with 46% F1 score and 82.9% AUC
- The LAHST outperforms state-of-the-art models such as HTDS and PubMedBERT-Hier across various early prediction settings, particularly excelling in scenarios where discharge summaries are not yet available
- The model incorporates causal attention mechanisms to ensure predictions are based solely on past data, making it ideal for real-time clinical decision support
- Training is optimized using the Extended Context Algorithm (ECA), which enhances memory efficiency and allows for longer EHR sequences to be included during inference, improving performance

FRONTEND DISPLAY

We integrated graphical insights on the frontend with the patient dashboard through dynamic and interactive visualizations. These visualizations will effectively represent model predictions, data trends, and key healthcare insights.

Designed to highlight critical insights for decision-making, such as patient outcomes, predictive accuracy, and model performance over time.





LEARNING OUTCOMES

This semester, our team gained valuable experience in integrating healthcare and machine learning technologies into the OpenEMR platform. We expanded our technical skills in debugging, database configuration, and system setup, deepening our understanding of healthcare workflows and predictive modeling. Key learnings included:

- Integration & Deployment: We worked with REST APIs, Docker, and Kubernetes to containerize and deploy machine learning models, strengthening our full-stack and infrastructure skills.
- Database Analysis: We identified critical components of healthcare databases, understanding their role in clinical and machine learning applications.
- Research & Model Application: We conducted literature reviews and explored open-source machine learning models to identify and implement suitable solutions for healthcare predictions.

FUTURE DIRECTIONS

- API-Frontend Integration: Connect the frontend to the Flask API by implementing functionality to send user inputs to the API and display the returned model predictions. This includes designing the frontend components to handle asynchronous requests and responses.
- Interactive Display: Enhance the frontend to dynamically update visualizations, such as graphs and charts, based on the API responses. Implement features for interactive data exploration, ensuring a responsive and user-friendly experience.

CITATIONS
[Continuous Predictive Modeling of Clinical Notes and ICD Codes in Patient Health Records] (https://aclanthology.org/2024.bionlp-1.19) (Caralt et al., BioNLP-WS 2024)