MEDS-TAB: SCALABLE TABULARIZATION FOR REPRODUCIBLE MEDS BASELINES

Anonymous authors
Paper under double-blind review

ABSTRACT

Classical tabular models are widely recognized as strong, often competitive baselines for structured prediction tasks on electronic health record (EHR) data, making them an important evaluation target for new machine learning models in this space. However, producing such baselines for longitudinal EHR datasets remains difficult: before any downstream model can be trained, researchers must convert irregular longitudinal event streams into temporal leakage-safe, task-aligned, fixed-dimensional feature matrices. While modern AutoML and hyperparameter optimization tools are incredibly effective at tuning models once such a matrix exists, they largely assume that this upstream tabularization problem has already been solved; however, this is often not the case. To help address this challenge, we introduce MEDS-Tab, a scalable tabularization system for datasets represented in the Medical Event Data Standard (MEDS). Given a MEDS event-stream-formatted dataset and task labels, MEDS-Tab constructs sparse, sharded feature matrices over heterogeneous clinical codes, lookback windows, and aggregation functions, with optional utilities for training downstream tabular baselines. MEDS-Tab scales to datasets with hundreds of millions of events and tens of thousands of codes. In our scalability experiments, MEDS-Tab completes all runs, while competing tools take much longer, and use substantially more memory or run out of memory. MEDS-Tab is also usable as shared infrastructure: it provides public documentation and a MIMIC-IV tutorial. It has received 17,836 PyPI downloads as of September 25, 2026 and has been used to generate baselines in five external studies. Across these studies, MEDS-Tab provides strong tabular baselines, achieving higher reported AUROC than every conventional baseline in 90.2% of study–dataset–task settings and outperforms all models, including neural and foundation models, in 62.7% of settings. By filling the infrastructure gap between standardized EHR event streams and reproducible classical ML baselines, MEDS-Tab improves the ease, scalability, and reproducibility of benchmarking across clinical prediction tasks and publications.

1 INTRODUCTION

Tabular baseline methods, particularly those powered by XGBoost (6), remain the gold standard for structured and longitudinal medical data (55; 34; 28; 33), serving as competitive baselines on many clinical tasks. A new sequence model or foundation architecture is only as convincing as its performance relative to a strong, reproducible classical baseline. Yet, despite their importance, producing these baselines remains a significant point of friction for researchers.

Currently, the "tabularization" process needed to produce these baselines is a manual, repetitive bottleneck. Longitudinal EHR data (which consists of irregularly sampled event streams, often with tens of thousands of unique codes) must be converted into fixed-dimensional feature matrices aligned to specific prediction times. Because these pipelines are often rebuilt from scratch for every new dataset and task, a massive amount of research effort is wasted on functionally identical work. More importantly, this reliance on bespoke pipelines undermines the robustness of the field: comparison results become implementation-dependent, making it difficult to discern whether a model's performance stems from its architecture or the idiosyncratic tuning of its baseline.

The medical ML community is in need of a reusable infrastructure that bridges the gap between raw data and modeling. While modern AutoML tools can tune a model once a feature matrix exists

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

(56; 52), they do not solve the upstream challenge of constructing that matrix in a temporal leakage-safe, scalable, and task-aligned way. This gap is especially severe for EHR datasets with tens of thousands of clinical codes and hundreds of millions of events, where dense wide-table approaches become impractical.

To address this challenge, we introduce MEDS-Tab, a scalable tabularization system for datasets represented in the Medical Event Data Standard (MEDS) (3). MEDS-Tab converts MEDS event streams and task labels into sparse, sharded, temporal leakage-safe feature matrices that can be consumed by XGBoost, Scikit-Learn models, or external sparse-matrix pipelines. The system includes downstream modeling utilities for convenience, but the central contribution is the reusable transformation from standardized longitudinal event data to task-aligned tabular representations.

MEDS-Tab is intended to make classical tabular baselines easier to construct, reuse, and compare for MEDS-based EHR prediction tasks. Given a MEDS dataset, task labels, prediction times, and train/tuning/held-out splits, MEDS-Tab produces a standardized sparse feature matrix for the same task definition and prediction times. This allows new models to be compared against a reproducible tabular baseline without requiring each study to rebuild its own feature-engineering pipeline. MEDS-Tab does not determine whether a task is clinically meaningful, whether labels were extracted correctly, or whether a model is suitable for deployment. Its role is to standardize the feature-construction step so that baseline comparisons are easier to reproduce and interpret across datasets and publications.

We make three contributions. First, we present MEDS-Tab, a scalable tabularization system for MEDS datasets that converts event shards and task labels into sparse, sharded, temporal leakage-safe feature matrices at user-specified prediction times. Second, we provide a practical baseline workflow, including command-line tools, public documentation, a MIMIC-IV tutorial, a usage guide, and optional XGBoost and Scikit-Learn training utilities. Third, we evaluate MEDS-Tab through scalability experiments and evidence of external adoption: MEDS-Tab had received 17,836 PyPI downloads as of September 25, 2026. Across five external studies, MEDS-Tab provides strong tabular baselines, achieving higher reported AUROC than every conventional baseline in 90.2% of study–dataset–task settings and outperforms all models, including neural and foundation models, in 62.7% of settings. (15; 16; 13; 17; 19).

2 Background and Problem Setting

2.1 Tabularization of EHR data

Structured EHR datasets are often described as tabular, but longitudinal EHR records are usually event streams: each row records that a subject had a code, value, or other observation at a particular time. Classical tabular models such as XGBoost (6) instead require a fixed-dimensional row for each prediction instance. A tabular baseline therefore requires a task-aligned feature vector that summarizes each subject's history before a prediction time.

We use *tabularization* to refer to this conversion from irregular event streams to fixed-dimensional feature matrices. Let a MEDS event be $e = (s, t, c, v)$, where s is a subject, t is a timestamp, c is a clinical code, and v is an optional numerical value. A supervised task provides labels (s_i, τ_i, y_i), where τ_i is the prediction time. For each label, a tabularizer constructs features using events up to τ_i. For code c, aggregation function g, and lookback window w, a numerical feature can be written as

$$x_{i,c,g,w} = g\left(\{e : \ e = (s_i, t, c, v_e),\ \tau_i - w \leq t \leq \tau_i\}\right).$$

Thus each row is indexed by a task label, each column is indexed by a code–aggregation–window triple, and temporal leakage safety is enforced by excluding observations after τ_i. This simple expression hides the engineering burden that MEDS-Tab addresses: the feature space is the Cartesian product of codes, windows, and aggregations, and the rows must be generated only at valid prediction times without materializing dense event-time matrices.

2.2 Before MEDS-Tab, creating tabular baselines was hard

We identify three major challenges to extracting tabular features at scale for EHR datasets. First and foremost, hospitals have **variable EHR schemas**, i.e., different hospitals use different medical code names in their systems. Several past papers implement dataset-specific tabularization pipelines for

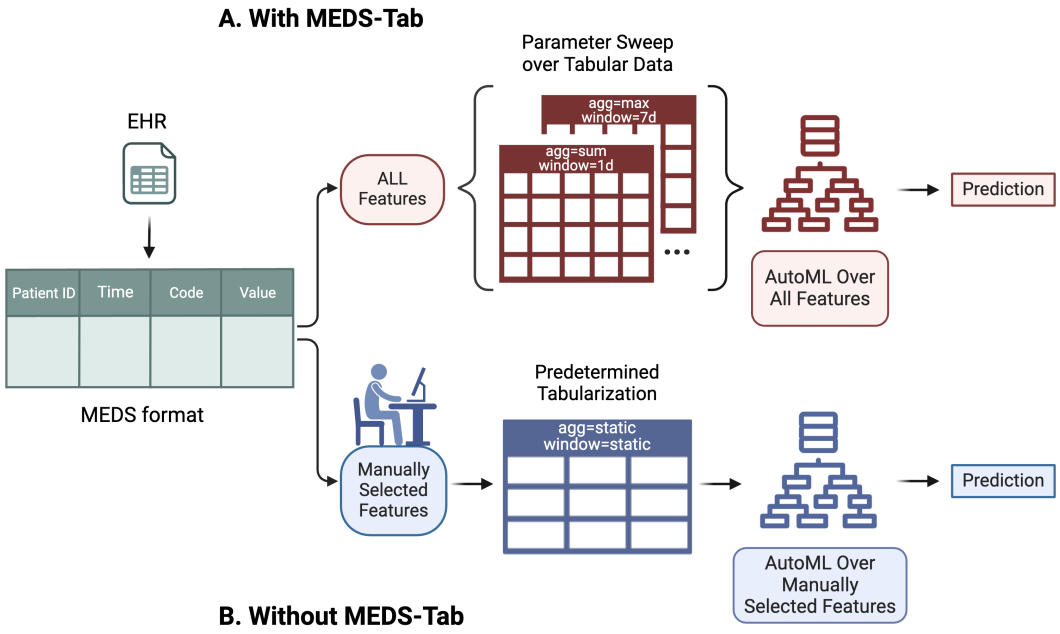

Figure 1: **MEDS-Tab workflow.** MEDS-Tab converts standardized MEDS event streams and task labels into sparse, task-aligned feature matrices that can be used by standard downstream baseline models. Without MEDS-Tab, researchers often rebuild dataset-specific feature construction, temporal windowing, temporal leakage checks, and baseline pipelines for each study.

the medical codes in the dataset they publish results for; however, those pipelines do not transfer to EHR systems with different medical codes (28; 68; 34). Alternatively, the OMOP EHR data format standardizes EHR data using a predefined schema and controlled concept vocabularies. OMOP-based pipelines (such as (33)) generally require users to manually define feature-extraction logic and assume that the source data have already been fully mapped to OMOP concepts. Consequently, applying them to a new or arbitrary EHR dataset first requires substantial harmonization and vocabulary-mapping effort.

Second, **supporting tabularization at scale** on datasets such as MIMIC-IV (30), a commonly used public dataset, presents substantial computational challenges. In EHR settings with hundreds of thousands of patients, tens of thousands of codes, and multiple prediction times per patient, generating a feature matrix easily produces over a billion entries, even before accounting for diverse aggregation functions and lookback windows. A naive implementation can require prohibitive memory and wall time (5). To our knowledge, two commonly used existing libraries (Catabra and tsfresh (40; 9)) support tabularization of longitudinal data, and we show that neither scales to the large code-set setting prevalent in EHR data.

Finally, the third challenge is **minimizing the implementation burden** for end users. While existing libraries such as CaTabRa (40) and tsfresh (9) provide helper functions for tabularization, they lack end-to-end workflows to go from raw data to fitted prediction models on tabularized data. Users must manually format inputs into standardized structures, explicitly supply clinical codes alongside patient-level start and end dates, handle feature storage, and manage data loading for downstream models. Crucially, users are also tasked with preventing temporal data leakage by manually guaranteeing that features are restricted to time windows prior to each prediction timestamp. This introduces substantial engineering overhead.

To address these three key challenges, we introduce MEDS-Tab: a framework designed to efficiently construct high-dimensional feature matrices across arbitrary EHR datasets with minimal user friction. In the following section, we outline the design of MEDS-Tab and detail how it resolves these computational and usability bottlenecks.

3 MEDS-TAB MAKES TABULARIZATION EASIER

1. **Standardized Input Interfaces:** By building natively on the Medical Event Data Standard (MEDS) schema, MEDS-Tab decouples tabularization from specific dataset implementations, making the pipeline inherently **dataset-agnostic**.
2. **Sparse, Sharded Data Pipelines:** To achieve **scalability**, MEDS-Tab avoids dense intermediate representations by utilizing sparse feature matrices and partitioning computations across parallelized data shards.
3. **End-to-End CLI Workflows:** To reduce **engineering burden**, MEDS-Tab provides modular, pre-configured command-line utilities that handle the entire lifecycle—from tabularization of MEDS event streams to downstream feature loading.

3.1 DATASET-AGNOSTIC

To address the challenge of **variable EHR schemas** across hospitals, remove the barrier of harmonizing medical codes, and help users avoid having to reimplement tabularization for each paper and dataset, MEDS-Tab relies on EHR data being in the MEDS format. Recent work (18) has documented the growing adoption of MEDS and the availability of public, well-maintained ETLs for most publicly available EHR datasets, as well as turnkey ETLs for any dataset already in OMOP CDM or FHIR format. A key design advantage of MEDS is its minimal schema, which makes onboarding a new dataset comparatively lightweight "without mandating deep harmonization of clinical vocabularies" (18). Additionally, MEDS-Tab does not require users to define a subset of codes to tabularize, and can generate features over the full medical codeset. This design choice reduces the barrier to obtaining baseline tabular model predictions from MEDS-Tab on a new dataset.

3.2 SCALABLE

MEDS-Tab supports tabularization at the scale of common EHR datasets by leveraging EHR data sparsity. Because individual clinical codes appear infrequently across patient histories in a typical EHR dataset, constructing a sparse feature matrix–rather than a dense matrix–dramatically reduces memory overhead. Combined with data sharding, this approach allows feature matrix generation to be parallelized across many processes on the same machine or across multiple nodes. As a result, MEDS-Tab can construct massive feature sets and reduces the need to aggressively drop potentially informative codes for computational efficiency.

3.3 STREAMLINING

MEDS-Tab streamlines end-to-end feature extraction by tabularizing data directly at task prediction times. Users simply provide a task dataframe (containing subject IDs, prediction timestamps, and labels) alongside their MEDS dataset directory to an entrypoint script (e.g., `MIMICIV_TUTORIAL/task_tabularize_meds.sh`). MEDS-Tab automates the entire pipeline through downstream model evaluation. It manages feature storage and loading across multiple lookback windows and aggregation functions, providing a default Optuna hyperparameter search with XGBoost for immediate benchmarking. Features are computed across lookback windows ending at the prediction time—defaulting to $1, 7, 30, 365$, and ∞ days—with all supported aggregation functions applied. Users can easily customize these windows via command-line arguments. Crucially, because aggregations are strictly computed backwards from prediction times, MEDS-Tab excludes time-stamped observations after the supplied prediction time, so users do not need to implement this filtering themselves.

4 MEDS-TAB SYSTEM

MEDS-Tab takes a MEDS dataset and task labels specifying a subject, prediction time, and outcome, and produces sparse feature matrices with one row per task label. The stages below follow the order of execution; Appendix A provides the commands.

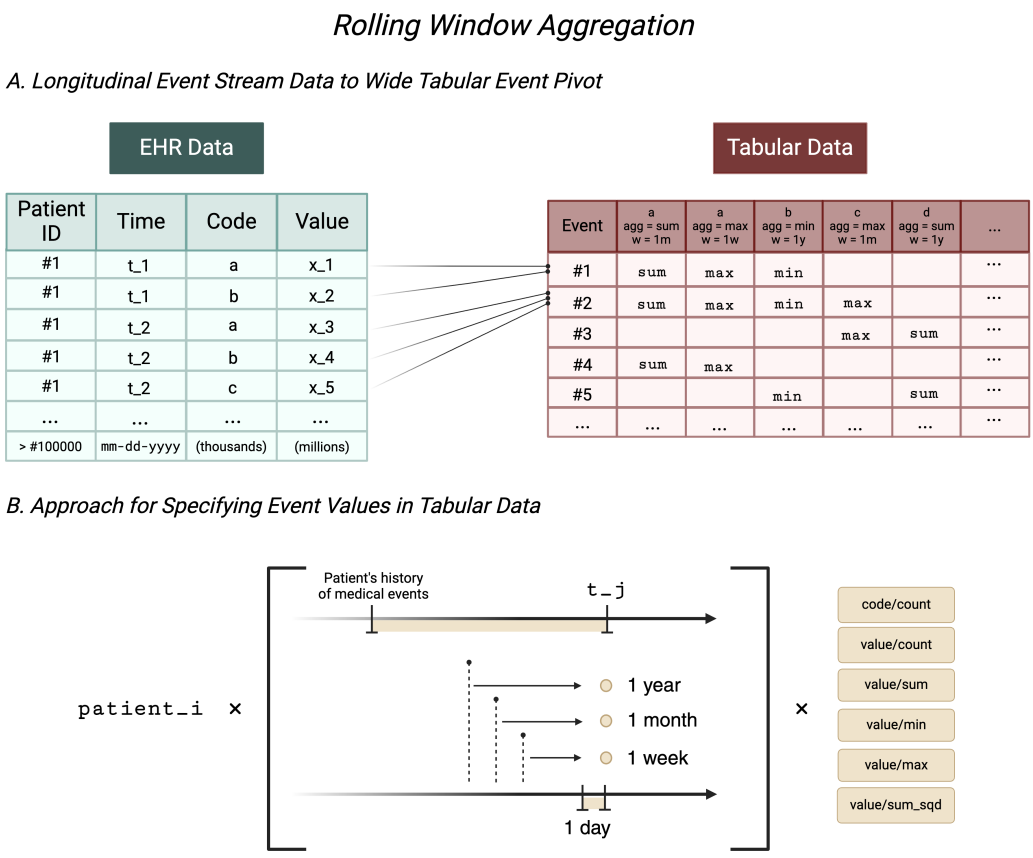

Figure 2: **EHR tabularization.** MEDS-Tab aggregates codes and numerical values over lookback windows to produce feature vectors for the supplied task labels.

4.1 Data sharding

A MEDS dataset is stored in Parquet files called *shards*, each containing all observations for a subset of subjects. Task labels follow the same split and shard organization. Keeping each subject's history together allows each shard to be tabularized independently, without combining observations from other shards. Users can optionally create smaller shards via the reshard CLI command to reduce the data processed by each worker (effectively reducing per-worker memory usage).

4.2 Code description

The first stage counts code occurrences, distinguishing static codes (codes with no timestamp, such as sex) from time-series codes (codes with timestamps). Counts are computed within each shard and combined into a shared metadata table. Tabularization uses a sorted feature list derived from this table, so each feature occupies the same column across shards. The counts also allow users to exclude rare codes in later steps, if desired.

4.3 Static and time-series tabularization

Static features. Static MEDS codes have no timestamp. MEDS-Tab supports two features per static code: a presence indicator (1 if the code is present for the subject, 0 otherwise) and its numeric value, when available. These features are computed for each shard and stored on disk as sparse matrices.

Time-series features. MEDS-Tab uses observations sorted by subject and timestamp to construct a sparse observation matrix with one row per observation and one column per code. It supports

six aggregations per code: the number of occurrences, the number of occurrences with a non-null numeric value, and the sum, sum of squares, minimum, and maximum of those values. The resulting features are stored on disk as sparse matrices.

After grouping observations by subject and timestamp, MEDS-Tab identifies the rows in the sparse observation matrix corresponding to each lookback window for each prediction time. MEDS-Tab aggregates those rows using sparse matrix operations. Unlike CaTabRa and tsfresh, MEDS-Tab does not copy history for each window, reducing runtime and memory use. The final matrix contains one feature row per task label (Figure 2).

4.4 Parallelization and storage

Static tabularization is quite fast since few codes are generally static in EHR datasets. Temporal tabularization, on the other hand, is divided into jobs, each processing one shard for one lookback window and aggregation. Workers process these jobs independently and write each result to a separate sparse `.npz` file. File locks prevent concurrent writes to the same output, and existing outputs are skipped when overwriting is disabled. This allows multiple workers to tabularize the data without loading the full dataset into memory.

4.5 Model training

For each shard, the loader concatenates the selected static and time-series matrices column-wise and pairs them with the task labels. It can then combine rows from multiple shards. MEDS-Tab provides XGBoost and Scikit-Learn training utilities, with optional Optuna searches over model parameters and feature choices (1). Users can also load the cached sparse matrices into their own training code and fit their own preferred model.

5 Efficiency Gains of MEDS-Tab

Table 1: **Scalability stress test for tabularization.** Wall time and memory for code-count aggregation over full subject histories up to every event time. Runs used an Intel Xeon Gold 5418Y CPU and 512 GB RAM. Times are reported as h:mm:ss and memory in GB. OOM indicates an out-of-memory failure; starred runtimes indicate elapsed time before OOM failure. Gray rows highlight MEDS-Tab. Best results among completed runs for each dataset, subject count, and metric are in bold.

		10 subjects			**100 subjects**			**500 subjects**		
Dataset	Method	Time	Avg GB	Peak GB	Time	Avg GB	Peak GB	Time	Avg GB	Peak GB
eICU	**MEDS-Tab (Ours)**	**0:00:06**	**5.02**	**13.04**	**0:00:30**	**9.74**	**13.93**	**0:02:19**	**10.93**	**20.33**
	tsfresh	0:15:27	138.66	313.09	0:09:10*	OOM	OOM	1:12:13*	OOM	OOM
	CaTabRa	0:11:00	80.25	230.05	9:29:02*	OOM	OOM	1:06:11*	OOM	OOM
MIMIC-IV	**MEDS-Tab (Ours)**	**0:00:03**	**0.38**	**1.50**	**0:00:05**	**0.59**	**1.58**	**0:00:14**	**1.42**	**5.07**
	tsfresh	0:01:15	8.34	21.34	0:04:24	15.82	35.59	0:13:13	25.58	59.45
	CaTabRa	0:00:31	1.32	4.40	0:03:11	13.64	27.36	0:24:43	23.75	62.56

We evaluate the computational scalability of MEDS-Tab against tsfresh and CaTabRa on MIMIC-IV and eICU. For each dataset, we select subsets of $n \in \{10, 100, 500\}$ subjects and retain all medical codes without filtering. At every event time, we compute the count of each code over the subject's full observed history up to that time, corresponding to a single aggregation function (`code/count`) and a full history lookback window. All methods run on an Intel Xeon Gold 5418Y CPU with 512 GB RAM until completion or an out-of-memory (OOM) error, with no timeout. We report wall time and average and peak memory usage; starred runtimes indicate elapsed time before OOM rather than time to completion.

MEDS-Tab completes all six evaluated settings and achieves the lowest runtime and average and peak memory usage among completed runs (Table 1). On MIMIC-IV, all three methods complete every setting, but their computational requirements differ substantially. For 500 subjects, MEDS-Tab finishes in 14 seconds, compared with 13 minutes 13 seconds for tsfresh and 24 minutes 43 seconds for CaTabRa, corresponding to approximately $57\times$ and $106\times$ speedups, respectively. Peak memory

usage is also substantially lower: 5.07 GB for MEDS-Tab versus 59.45 GB for tsfresh and 62.56 GB for CaTabRa.

On eICU, both tsfresh and CaTabRa run out of memory at 100 and 500 subjects, whereas MEDS-Tab completes the 500-subject setting in 2 minutes 19 seconds with 20.33 GB peak memory usage. Even for 10 subjects, where all methods complete, tsfresh and CaTabRa require 15 minutes 27 seconds and 11 minutes, respectively, compared with 6 seconds for MEDS-Tab. Their peak memory usage reaches 313.09 GB and 230.05 GB, compared with 13.04 GB for MEDS-Tab.

During tabularization, CaTabRa pairs each subject's events with every lookback window, filters events outside each window, and then aggregates by window and code. tsfresh similarly duplicates all history for each subject-code pair within every lookback window before applying group-by aggregations. It is this duplication of events, coupled with the dense output data formats in both prior approaches that leads to inflated memory and computational costs.

These results demonstrate the practical benefit of sparse tabularization on EHR datasets with many codes but high sparsity. MEDS-Tab substantially reduces runtime and memory usage in settings where all methods complete, while also completing workloads that exceed the available memory for both dense alternatives.

6 Community Adoption and External Benchmarking

This section discusses the rising adoption of MEDS-Tab and demonstrates that MEDS-Tab remains a strong baseline and a competitive predictor when compared against neural and foundation models across multiple recent studies.

6.1 PyPI usage and independent adoption

MEDS-Tab has emerged as a useful baselining protocol within the Medical Event Data Standard (MEDS) ecosystem. The package had received 17,836 PyPI downloads as of September 25, 2026, and has been used in five independent MEDS-based studies: Study 1, Study 2, Study 3, Study 4, and Study 5[1]. These uses are important because they test whether MEDS-Tab functions as reusable infrastructure outside the experiments designed for this paper.

Table 2: **Community adoption and external competitiveness of MEDS-Tab.** Each comparison is one study–dataset–task setting. Win rates give the percentage of settings in which MEDS-Tab's reported AUROC for a comparison strictly exceeds every other model for *All Models* or every other conventional baseline subset of models for *Baselines*. *# Models* provides the number of compared to models (excluding MEDS-Tab) and counts each compared to model once per study across the selected tasks. *# Neural Models* counts the total neural models compared against for the All models comparisons, and *# Baselines* counts the number of baselines compared to. The All row pools wins over 83 *All Models* and 61 *Baselines* comparisons and sums model counts across studies.

External Study	All Models	Baselines	# Neural Models	# Models	# Baselines	# Comparisons
Study 1	75.0%	100.0%	3	5	2	4
Study 2	14.3%	78.6%	7	9	2	14
Study 3	0.0%	66.7%	4	13	4	3
Study 4	81.8%	—	1	1	0	22
Study 5	72.5%	95.0%	8	9	1	40
All	**62.7%**	**90.2%**	**23**	**37**	**9**	**83**

[1]Citations to the five studies are redacted for review because each cites MEDS-Tab by name and would reveal the authors' identities. All five are published or publicly available (one ML4H 2024 paper, one GigaScience 2025 paper, and three arXiv preprints from 2025–2026), and full citations will be restored in the camera-ready version. All numerical results are reproduced verbatim from the source tables in Appendix E. Full citations are available upon request by the Area Chair.

6.2 COMPETITIVE PERFORMANCE ACROSS EXTERNAL STUDIES

Evaluation setup. To evaluate whether MEDS-Tab provides strong predictive baselines, we aggregate externally reported AUROC results from five studies: Study 1, Study 2, Study 3, Study 4, and Study 5 (13; 15; 16; 17; 19). These studies evaluate models trained on MEDS-Tab features alongside conventional tabular approaches, neural sequence models, and foundation models under their respective experimental protocols.

Table 2 contains 83 comparisons, each corresponding to one study–dataset–task setting. We consider two groups of models we evaluate MEDS-Tab against for each comparison: *Baselines*, comprising conventional baseline tabular or neural models, and *All Models*, comprising all non-MEDS-Tab predictors in the comparison. A win requires MEDS-Tab's reported AUROC to strictly exceed that of every other model in the group. Each study–dataset–task setting contributes one comparison regardless of the number of models evaluated. The model-count columns count each model MEDS-Tab was evaluated against once per study, and the pooled row sums wins, comparisons, and model counts across studies. Appendix E provides the source results, and *Baseline* group definitions for each study.

Strong conventional tabular baselines. MEDS-Tab outperforms every conventional tabular baseline in 55 of 61 comparisons (90.2%) across the four studies with conventional baselines. Its within-study win rate ranges from 66.7% to 100.0%. These comparisons include age-and-code-count representations in Study 1 and Study 2, as well as clinically selected emergency-department features in Study 3. Thus in these experiments, MEDS-Tab's automatically constructed features support strong predictive baselines relative to both simple tabularization approaches and task-specific, manually engineered representations.

Competitive with neural and foundation models. MEDS-Tab also achieves the highest reported AUROC among all models in 52 of 83 comparisons (62.7%). The *All Models* group includes between one and eight neural models per study, including foundation models. For example, in Study 5, MEDS-Tab outperforms all nine models, including eight neural models, in 29 of 40 settings. Four of those models are foundation-model variants (including both fine-tuned and linear-probed variants). In Study 1, it outperforms all five models, including three neural models, in three of four settings. Although the strongest predictor varies across studies and tasks, these results show that MEDS-Tab is competitive beyond the conventional baseline group. Together, the external evaluations support its use as a strong, reusable tabular baseline against which to assess advances in neural EHR prediction.

7 RELATED WORK

MEDS-Tab builds on several lines of work. Large-scale EHR resources and benchmark datasets have enabled broad clinical prediction research (29; 51; 20; 49; 62; 61), but the high-dimensional and sparsely measured structure of EHR data remains a central modeling challenge (27; 65; 66). Time-series tabularization libraries such as tsfresh and sktime (9; 39), AutoML and hyperparameter optimization systems such as AutoGluon and Optuna (11; 1), and integrated clinical ML pipelines such as TemporAI and CaTabRa (54; 40) provide important components for this space.

The distinction is that MEDS-Tab targets the missing infrastructure layer between standardized EHR event streams and reusable tabular baselines. Many existing systems rely on dense *pandas.DataFrame* or *numpy.ndarray* containers, which become impractical when featurizing over many medical codes, aggregation functions, and lookback windows. Clinical deep learning toolkits such as PyHealth (69) support sequential model inputs, but they do not primarily address the generation of sparse, temporal leakage-safe, task-aligned 2D matrices for classical baselines. AutoML systems are complementary to this goal: they can tune downstream models once a matrix exists, whereas MEDS-Tab focuses on producing the sparse task-aligned matrices themselves. Appendix B provides a longer comparison.

8 LIMITATIONS

While MEDS-Tab streamlines the generation of reproducible baselines, it has several limitations. First, by design, MEDS-Tab tabularizes longitudinal event streams into aggregated, discrete time

windows. While appropriate for classical machine learning architectures such as XGBoost, this flattening process discards some fine-grained temporal information that sequence models may exploit.

Second, MEDS-Tab can enforce temporal leakage safety only with respect to the event and prediction times it is given. Static MEDS measurements are treated as time-invariant subject attributes. MEDS-Tab cannot determine whether a task definition is clinically meaningful, whether labels were extracted correctly, or whether upstream MEDS conversion preserved all relevant information. These issues remain the responsibility of dataset and task authors.

Third, MEDS-Tab currently focuses on structured EHR events. Integrating high-dimensional unstructured modalities, such as clinical text or continuous waveform signals, remains outside its current scope.

Fourth, while sparse matrices and sharding drastically reduce memory overhead compared to naive dense representations, extracting tens of thousands of features across hundreds of millions of events still imposes a non-trivial computational footprint. Some optional downstream utilities, including imputation-heavy Scikit-Learn workflows and AutoGluon, may densify data and therefore do not inherit the same scalability properties as the sparse tabularization and XGBoost paths.

Finally, MEDS-Tab requires MEDS format datasets. Converting a dataset to MEDS is relatively simple, and adoption of MEDS format is growing (18; 14), with multiple datasets publicly available in MEDS format and multiple tools aiding in this preprocessing step (18; 3); nonetheless, dependence on this format is a limitation.

9 Conclusion

MEDS-Tab is not a predictive model itself. It is scalable infrastructure for resolving the tabularization bottleneck that makes classical healthcare ML baselines hard to reproduce. By bridging the standardized MEDS format with sparse, sharded, temporal leakage-safe feature construction, MEDS-Tab allows researchers to bypass the traditionally manual and memory-prohibitive process of EHR feature engineering. The result is a reusable path from longitudinal clinical event streams to competitive tabular baselines for evaluating emerging foundation models and complex clinical architectures.

AI use statement

In this work, we did not use generative AI tools for any tasks with required disclosure. We used generative AI tools for drafting various portions of the text, improving existing text, and code review. We reviewed all AI-assisted work, including carefully reading and editing all AI-generated text and suggestions. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

Ethics statement

This work uses existing, publicly available, de-identified EHR datasets: MIMIC-IV (30) and the eICU Collaborative Research Database (51), both hosted on PhysioNet. All patient data in these repositories were de-identified prior to release.

Original data collection was approved by the Institutional Review Boards (IRBs) of the respective source institutions (e.g., Beth Israel Deaconess Medical Center and the Massachusetts Institute of Technology). The use of these secondary, fully de-identified datasets is exempt from further institutional IRB review. All researchers involved in this project completed the required CITI Program training in human subjects research and signed the PhysioNet Data Use Agreement. We strictly adhered to all DUA stipulations, including making no attempts to re-identify individuals or redistribute raw data. Furthermore, MEDS-Tab processes and transforms data locally.

Reproducibility statement

To ensure complete reproducibility of our work, we provide the full, open-source `MEDS-Tab` codebase, command-line interface (CLI), and public documentation in an anonymous repository. Detailed descriptions of data preprocessing, MEDS event stream parsing, and tabular feature extraction

pipelines are provided in the Appendix and online documentation. Furthermore, comprehensive step-by-step tutorials, execution scripts, and interactive notebooks required to reproduce all tabular benchmarks from raw MEDS data are included within the supplementary package.

REFERENCES

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

[2] Sarah Alnegheimish, Najat Alrashed, Faisal Aleissa, Shahad Althobaiti, Dongyu Liu, Mansour Alsaleh, and Kalyan Veeramachaneni. Cardea: An open automated machine learning framework for electronic health records. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 536–545. IEEE, 2020.

[3] Bert Arnrich, Edward Choi, Jason Alan Fries, Matthew B.A. McDermott, Jungwoo Oh, Tom Pollard, Nigam Shah, Ethan Steinberg, Michael Wornow, and Robin van de Water. Medical event data standard (MEDS): Facilitating machine learning for health. In *ICLR 2024 Workshop on Learning from Time Series For Health*, 2024.

[4] James Bergstra, Brent Komer, Chris Eliasmith, Dan Yamins, and David D Cox. Hyperopt: a python library for model selection and hyperparameter optimization. *Computational Science & Discovery*, 8(1):014008, 2015.

[5] Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[6] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[7] Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Coffey, Michael Thompson, James Bost, Javier Tejedor-Sojo, and Jimeng Sun. Multi-layer representation learning for medical concepts. In *proceedings of the 22nd ACM SIGKDD international conference on Knowledge Discovery and data mining*, pages 1495–1504, 2016.

[8] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. *Advances in neural information processing systems*, 29, 2016.

[9] Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W Kempa-Liehr. Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package). *Neurocomputing*, 307:72–77, 2018.

[10] Matthew M Churpek, Trevor C Yuen, Seo Young Park, David O Meltzer, Jesse B Hall, and Dana P Edelson. Derivation of a cardiac arrest prediction model using ward vital signs. *Critical care medicine*, 40(7):2102, 2012.

[11] Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. Autogluon-tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505*, 2020.

[12] Matthias Feurer, Katharina Eggensperger, Stefan Falkner, Marius Lindauer, and Frank Hutter. Auto-sklearn 2.0: The next generation. *arXiv preprint arXiv:2007.04074*, 24:8, 2020.

[13] Redacted for Anonymity. Study 1. *Machine Learning For Health Conference*, 2024.

[14] Redacted for Anonymity. Redacted for anonymity. ACM Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery, 2025.

[15] Redacted for Anonymity. Study 2. *arXiv preprint*, 2025.

[16] Redacted for Anonymity. Study 3. *GigaScience*, 2025.

[17] Redacted for Anonymity. Study 4. *arXiv preprint*, 2025.

[18] Redacted for Anonymity. Redacted for anonymity. *NEJM AI*, 2026.

[19] Redacted for Anonymity. Study 5. *arXiv preprint*, 2026.

[20] Tianfan Fu, Kexin Huang, Cao Xiao, Lucas M Glass, and Jimeng Sun. Hint: Hierarchical interaction network for clinical-trial-outcome predictions. *Patterns*, 3(4), 2022.

[21] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[22] Hrayr Harutyunyan, Hrant Khachatrian, David C Kale, Greg Ver Steeg, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific data*, 6(1):96, 2019.

[23] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[24] Kyunghoon Hur, Jungwoo Oh, Junu Kim, Jiyoun Kim, Min Jae Lee, Eunbyeol Cho, Seong-Eun Moon, Young-Hak Kim, Louis Atallah, and Edward Choi. Genhpf: General healthcare predictive framework for multi-task multi-source learning. *IEEE Journal of Biomedical and Health Informatics*, 28(1):502–513, 2023.

[25] Daniel Jarrett, Bogdan C Cebere, Tennison Liu, Alicia Curth, and Mihaela van der Schaar. Hyperimpute: Generalized iterative imputation with automatic model selection. In *International Conference on Machine Learning*, pages 9916–9937. PMLR, 2022.

[26] Daniel Jarrett, Jinsung Yoon, Ioana Bica, Zhaozhi Qian, Ari Ercole, and Mihaela van der Schaar. Clairvoyance: A pipeline toolkit for medical time series. *arXiv preprint arXiv:2310.18688*, 2023.

[27] Peter B Jensen, Lars J Jensen, and Søren Brunak. Mining electronic health records: towards better research applications and clinical care. *Nature Reviews Genetics*, 13(6):395–405, 2012.

[28] Hyewon Jeong, Nassim Oufattole, Aparna Balagopalan, Matthew Mcdermott, Payal Chandak, Marzyeh Ghassemi, and Collin Stultz. Event-based contrastive learning for medical time series. *arXiv preprint arXiv:2312.10308*, 2023.

[29] Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. Mimic-iv. *PhysioNet. Available online at: https://physionet. org/content/mimiciv/1.0/(accessed August 23, 2021)*, pages 49–55, 2020.

[30] Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, et al. Mimic-iv, a freely accessible electronic health record dataset. *Scientific data*, 10(1):1, 2023.

[31] Alistair EW Johnson, Tom J Pollard, and Roger G Mark. Reproducibility in critical care: a mortality prediction case study. In *Machine learning for healthcare conference*, pages 361–376. PMLR, 2017.

[32] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[33] Rohan Kodialam, Rebecca Boiarsky, Justin Lim, Aditya Sai, Neil Dixit, and David Sontag. Deep contextual clinical prediction with reverse distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 249–258, 2021.

[34] Alex Labach, Aslesha Pokhrel, Xiao Shi Huang, Saba Zuberi, Seung Eun Yi, Maksims Volkovs, Tomi Poutanen, and Rahul G. Krishnan. Duett: Dual event time transformer for electronic health records. In Kaivalya Deshpande, Madalina Fiterau, Shalmali Joshi, Zachary Lipton, Rajesh Ranganath, Iñigo Urteaga, and Serene Yeung, editors, *Proceedings of the 8th Machine Learning for Healthcare Conference*, volume 219 of *Proceedings of Machine Learning Research*, pages 403–422. PMLR, 11–12 Aug 2023.

[35] Alex Labach, Aslesha Pokhrel, Xiao Shi Huang, Saba Zuberi, Seung Eun Yi, Maksims Volkovs, Tomi Poutanen, and Rahul G Krishnan. Duett: dual event time transformer for electronic health records. In *Machine learning for healthcare conference*, pages 403–422. PMLR, 2023.

[36] Jiang Li, Xiaowei S Yan, Durgesh Chaudhary, Venkatesh Avula, Satish Mudiganti, Hannah Husby, Shima Shahjouei, Ardavan Afshar, Walter F Stewart, Mohammed Yeasin, et al. Imputation of missing values for electronic health record laboratory data. *NPJ digital medicine*, 4(1):147, 2021.

[37] Steven Cheng-Xian Li and Benjamin Marlin. Learning from irregularly-sampled time series: A missing data perspective. In *International Conference on Machine Learning*, pages 5937–5946. PMLR, 2020.

[38] Yikuan Li, Shishir Rao, José Roberto Ayala Solares, Abdelaali Hassaine, Rema Ramakrishnan, Dexter Canoy, Yajie Zhu, Kazem Rahimi, and Gholamreza Salimi-Khorshidi. Behrt: transformer for electronic health records. *Scientific reports*, 10(1):7155, 2020.

[39] Markus Löning, Anthony Bagnall, Sajaysurya Ganesh, Viktor Kazakov, Jason Lines, and Franz J Király. sktime: A unified interface for machine learning with time series. *arXiv preprint arXiv:1909.07872*, 2019.

[40] Alexander Maletzky, Sophie Kaltenleithner, Philipp Moser, and Michael Giretzlehner. Catabra: Efficient analysis and predictive modeling of tabular data. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 57–68. Springer, 2023.

[41] Matthew McDermott. MEDS "Everything-is-code" Autoregressive Model.

[42] Matthew McDermott, Bret Nestor, Evan Kim, Wancong Zhang, Anna Goldenberg, Peter Szolovits, and Marzyeh Ghassemi. A comprehensive evaluation of multi-task learning and multi-task pre-training on ehr time-series data. *arXiv preprint arXiv:2007.10185*, 2020.

[43] Ann McGinley and Rupert M Pearse. A national early warning score for acutely ill patients, 2012.

[44] A Tuan Nguyen, Hyewon Jeong, Eunho Yang, and Sung Ju Hwang. Clinical risk prediction with temporal probabilistic asymmetric multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9081–9091, 2021.

[45] Mikkel Fruelund Odgaard, Kiril Vadimovic Klein, Martin Sillesen, Sanne Møller Thysen, Espen Jimenez-Solem, and Mads Nielsen. CORE-BEHRT: A carefully optimized and rigorously evaluated BEHRT. In Kaivalya Deshpande, Madalina Fiterau, Shalmali Joshi, Zachary Lipton, Rajesh Ranganath, and Iñigo Urteaga, editors, *Proceedings of the 9th Machine Learning for Healthcare Conference*, volume 252 of *Proceedings of Machine Learning Research*. PMLR, 16–17 Aug 2024.

[46] Thomas Olsson, Andreas Terént, and Lars Lind. Rapid emergency medicine score: a new prognostic tool for in-hospital mortality in nonsurgical emergency department patients. *Journal of internal medicine*, 255(5):579–587, 2004.

[47] Chao Pang, Xinzhuo Jiang, Krishna S Kalluri, Matthew Spotnitz, RuiJun Chen, Adler Perotte, and Karthik Natarajan. Cehr-bert: Incorporating temporal information from structured ehr data to improve prediction tasks. In *Machine learning for health*, pages 239–260. PMLR, 2021.

[48] Chao Pang, Xinzhuo Jiang, Nishanth Parameshwar Pavinkurve, Krishna S Kalluri, Elise L Minto, Jason Patterson, Linying Zhang, George Hripcsak, Gamze Gürsoy, Noémie Elhadad, et al. Cehr-gpt: Generating electronic health records with chronological patient timelines. *arXiv preprint arXiv:2402.04400*, 2024.

[49] Ke Pang, Liang Li, Wen Ouyang, Xing Liu, and Yongzhong Tang. Establishment of icu mortality risk prediction models with machine learning algorithm using mimic-iv database. *Diagnostics*, 12(5):1068, 2022.

[50] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[51] Tom J Pollard, Alistair E W Johnson, Jesse D Raffa, Leo A Celi, Roger G Mark, and Omar Badawi. The eICU Collaborative Research Database, a freely available multi-center database for critical care research. *Scientific data*, 5(1):1–13, 2018.

[52] Hooman H Rashidi, Nam Tran, Samer Albahra, and Luke T Dang. Machine learning in health care and laboratory medicine: General overview of supervised learning and auto-ml. *International Journal of Laboratory Hematology*, 43:15–22, 2021.

[53] Ruichen Rong, Zifan Gu, Hongyin Lai, Tanna L Nelson, Tony Keller, Clark Walker, Kevin W Jin, Catherine Chen, Ann Marie Navar, Ferdinand Velasco, et al. A deep learning model for clinical outcome prediction using longitudinal inpatient electronic health records. *JAMIA open*, 8(2):ooaf026, 2025.

[54] Evgeny S Saveliev and Mihaela van der Schaar. Temporai: Facilitating machine learning innovation in time domain tasks for medicine. *arXiv preprint arXiv:2301.12260*, 2023.

[55] Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. In *8th ICML Workshop on Automated Machine Learning (AutoML)*, 2021.

[56] Vivek Kumar Singh and Kailash Joshi. Automated machine learning (automl): an overview of opportunities for application and research. *Journal of Information Technology Case and Application Research*, 24(2):75–85, 2022.

[57] Gary B Smith, Oliver C Redfern, Marco AF Pimentel, Stephen Gerry, Gary S Collins, James Malycha, David Prytherch, Paul E Schmidt, and Peter J Watkinson. The national early warning score 2 (news2). *Clinical Medicine*, 19(3):260, 2019.

[58] Micah J Smith, Carles Sala, James Max Kanter, and Kalyan Veeramachaneni. The machine learning bazaar: Harnessing the ml ecosystem for effective system development. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 785–800, 2020.

[59] Ethan Steinberg, Jason Fries, Yizhe Xu, and Nigam Shah. Motor: A time-to-event foundation model for structured medical records. In *International Conference on Learning Representations*, volume 2024, pages 41038–41077, 2024.

[60] Christian P Subbe, Michael Kruger, Peter Rutherford, and L Gemmel. Validation of a modified early warning score in medical admissions. *Qjm*, 94(10):521–526, 2001.

[61] Wen Sun, Yang Yan, Shidong Hu, Boyan Liu, Shuying Wang, Wenli Yu, and Songyan Li. The effects of midazolam or propofol plus fentanyl on icu mortality: a retrospective study based on the mimic-iv database. *Annals of Translational Medicine*, 10(4), 2022.

[62] Yiwu Sun, Zhaoyi He, Jie Ren, and Yifan Wu. Prediction model of in-hospital mortality in intensive care unit patients with cardiac arrest: a retrospective analysis of mimic-iv database based on machine learning. *BMC anesthesiology*, 23(1):178, 2023.

[63] Sindhu Tipirneni and Chandan K Reddy. Self-supervised transformer for sparse and irregularly sampled multivariate clinical time-series. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(6):1–17, 2022.

[64] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[65] Nicole G Weiskopf, George Hripcsak, Sushmita Swaminathan, and Chunhua Weng. Defining and measuring completeness of electronic health records for secondary use. *Journal of biomedical informatics*, 46(5):830–836, 2013.

[66] Brian J Wells, Kevin M Chagin, Amy S Nowacki, and Michael W Kattan. Strategies for handling missing data in electronic health record derived data. *Egems*, 1(3), 2013.

[67] Richard C Wuerz, Leslie W Milne, David R Eitel, Debbie Travers, and Nicki Gilboy. Reliability and validity of a new five-level triage instrument. *Academic emergency medicine*, 7(3):236–242, 2000.

[68] Feng Xie, Jun Zhou, Jin Wee Lee, Mingrui Tan, Siqi Li, Logasan S/O Rajnthern, Marcel Lucas Chee, Bibhas Chakraborty, An-Kwok Ian Wong, Alon Dagan, et al. Benchmarking emergency department prediction models with machine learning and public electronic health records. *Scientific Data*, 9(1):658, 2022.

[69] Yue Zhao, Zhi Qiao, Cao Xiao, Lucas Glass, and Jimeng Sun. Pyhealth: A python library for health predictive models. *arXiv preprint arXiv:2101.04209*, 2021.

A Command Line Interface

The recommended MEDS-Tab workflow is task-specific tabularization. The MIMIC-IV tutorial wraps the core commands in `task_tabularize_meds.sh`, which takes a MEDS dataset directory, a task list, a task-label directory, output directories, and a worker count. For each task, it emits rows aligned to the prediction times present in the task labels and then runs a simple XGBoost sweep. The same script can be used on other MEDS-format datasets with compatible label files.

Optional Resharding (`MEDS_transform-reshard_to_split`) Before running MEDS-Tab, users can optionally reshard a MEDS dataset into smaller subject-level shards. A shard of MEDS data is a dataframe parquet on disk that contains all the observations for a subset of subjects. Shards are the unit of work during parallelization, as a worker will tabularize one shard at a time. This resharding command is useful when the original shards are too large for comfortable parallel processing or memory usage.

Data Description (`meds-tab-describe`) Analyzes MEDS data shards to compute code frequencies, categorizing them into time-series codes, static codes, and their numerical variants. Results are cached in `metadata/codes.parquet`. These frequencies can optionally be used to filter codes in later steps via arguments such as `tabularization.min_code_inclusion_frequency=X`.

Static Data Tabularization (`meds-tab-tabularize-static`) Transforms static subject data into sparse feature vectors. In the recommended task-specific workflow, this command is run with an `input_label_dir` argument, so static rows are emitted for the task's prediction times rather than for all event times in the dataset. Static MEDS measurements are treated as time-invariant subject attributes. Storing static features sparsely makes them easy to concatenate with time-series features during model training.

Time-Series Data Tabularization (`meds-tab-tabularize-time-series`) Generates temporal feature vectors by aggregating subject time-series data across the requested window sizes and aggregation methods (Figure 2). In the recommended workflow, the command also receives `input_label_dir`; MEDS-Tab then computes rolling windows directly at the task prediction times and writes the corresponding sparse matrices. This avoids the older, less efficient pattern of first tabularizing every subject event time and then filtering rows for a task. Data are pre-sorted by subject ids and timestamps, and Polars is used to compute rolling indices for the requested windows. Because shards are independent and outputs are sparse, this stage parallelizes naturally across workers while keeping memory overhead low.

Model Training (`meds-tab-xgboost` and `meds-tab-model`) The tutorial uses `meds-tab-xgboost` to train a simple XGBoost baseline over the task-specific tabularized matrices, with an Optuna sweep over model and featurization choices. The more general `meds-tab-model` command supports additional Scikit-Learn (50) classifiers. For datasets that do not fit in memory, supported models can load one shard at a time via `data_loading_params.keep_data_in_memory=False`.

B Extended Related Works

The existing body of work in medical time-series analysis can be broadly categorized into three main areas: the handling and tabularization of irregular temporal data, the automation of modeling processes through AutoML (56; 52), and the streamlining of pipelines to enhance efficiency. While foundational, these isolated solutions often face computational or modularity bottlenecks when applied to modern, large-scale EHR datasets.

Tabularization of Time-Series Data The process of converting irregularly sampled time-series data into a structured tabular format is essential for the effective application of machine learning models. Tools like sktime (39), tsfresh (9), Clairvoyance (26), and TemporAI (54) have significantly

advanced this area by offering robust frameworks for feature extraction and transformation. These tools enable tabularization by aggregating features across various time windows and handling missing values. For example, tsfresh automates the extraction of relevant features, while sktime provides a unified framework for time-series analysis. However, tsfresh states that the memory consumption of parallelized calculations can be high, making it infeasible on low-memory machines. In the healthcare domain, Clairvoyance (superseded by TemporAI) offers tabularization toolkits, but both systems use dense data containers like *pandas.DataFrame* or *numpy.ndarray* which do not handle the sparse nature of EHR data efficiently.

AutoML The automation of machine learning workflows has greatly simplified model development. Tools such as AutoGluon (11), Optuna (1), and hyperopt (4) exemplify this by automating elements of algorithm selection, hyperparameter tuning, and model evaluation. AutoGluon offers a comprehensive framework automating the end-to-end process of model selection and ensembling across various tasks. While powerful, the automation of complex processes like model stacking and ensembling can lead to significant demands on CPU, memory, and disk space, making it less suitable for computationally limited environments or lightweight model training. Conversely, Optuna has emerged as a flexible framework enabling fine-tuning of models with minimal manual intervention.

Integrated Pipelines Integrated pipelines aim to bridge the gap between tabularization and AutoML by combining their strengths within a cohesive framework. CaTabRa (40) offers a largely automated workflow integrating established libraries like auto-sklearn (12). However, its "automate what can reasonably be automated" approach means complex tasks, particularly defining and extracting meaningful target labels, still require significant user expertise. Cardea (2) is tailored specifically for EHR data, leveraging tools like ML-Bazaar (58) and hyperopt; however, its general applicability is limited by the predefined problem definitions it supports. Clairvoyance focuses on medical time series but has been critiqued for insufficient modularity and limited community support. TemporAI addresses these shortcomings by offering a more modular framework for temporal data, supporting various data modalities and causal inference. Nonetheless, like its predecessor, TemporAI's reliance on *pandas.DataFrame* requires significant computational resources.

Further Considerations Additional contributions in the field include encoding techniques (63) and robust modeling approaches like self-supervised learning (28; 37) and multi-task learning (22; 44; 42) to manage irregular sampling. The field has also seen significant improvements in imputation strategies to address data missingness (36; 25). In terms of architectural toolkits, PyHealth (69) provides access to over 30 state-of-the-art models for healthcare predictive modeling. However, PyHealth 2.0 has no dedicated tabularization method.

C Baseline performance on clinical prediction tasks

To illustrate the practical application of MEDS-Tab, we report XGBoost AUROCs for common clinical prediction tasks on MIMIC-IV and Philips eICU, including readmission, mortality, and length-of-stay prediction (Table 3). These results are not intended to introduce a new model class. They show that the automatically produced MEDS-Tab matrices support nontrivial classical baselines across tasks, datasets, and prediction times.

These results are best interpreted as a smoke test of the artifact's default workflow: with the same tabularization machinery, MEDS-Tab produces usable baselines across discharge, 24-hour, and 48-hour prediction settings without task-specific feature-engineering code.

D More details on comparisons

Study 2, Study 1, Study 3, and Study 5 compare MEDS-Tab against models trained on manually constructed tabular representations. We describe these comparisons below.

Study 2 directly compares two tabularization strategies across 14 tasks on the CUMC data. Its tabular baseline represents each patient using the following manual featurization: age and the count of occurrences of every code between the patient's first recorded event and the prediction time, followed

Table 3: **Default XGBoost baselines from MEDS-Tab features.** AUROC, subject counts, and task label row counts for MIMIC-IV and eICU clinical prediction tasks. Note that the Subjects (k) and Label rows (k) columns are reported in thousands (for example, an entry of 2 corresponds to 2,000).

Prediction Timestamp	Dataset	Task	AUC	Subjects (k)	Label rows (k)
Discharge	MIMIC-IV	30 day Mortality	0.935	149	356
		1 year Mortality	0.898	149	356
		30 day Readmission	0.708	17	378
Admission + 24 hr	MIMIC-IV	In ICU Mortality	0.661	8	23
		In Hospital Mortality	0.812	51	339
		LOS in ICU > 3 days	0.946	43	61
		LOS in Hospital > 3 days	0.943	152	360
	eICU	In Hospital Mortality	0.855	4	4
		LOS in ICU > 3 days	0.783	13	14
		LOS in Hospital > 3 days	0.864	100	100
Admission + 48 hr	MIMIC-IV	In ICU Mortality	0.673	7	21
		In Hospital Mortality	0.810	47	348
		LOS in ICU > 3 days	0.967	43	61
		LOS in Hospital > 3 days	0.945	152	359
	eICU	In Hospital Mortality	0.570	2	2
		LOS in ICU > 3 days	0.757	13	14
		LOS in Hospital > 3 days	0.895	100	100

by logistic regression or LightGBM. MEDS-Tab provides the strongest conventional baseline on 11 of these 14 tasks.

Study 1 similarly evaluates MEDS-Tab against logistic-regression and LightGBM baselines (using the same manual age + count featurization as Study 2) for predicting long length of stay and ICU Mortality on MIMIC-IV and on Columbia data. MEDS-Tab provides the strongest conventional result on all four evaluated tasks.

Study 3 evaluates MEDS-Tab on three MIMIC-IV-ED prediction tasks: hospitalization from the ED and a critical outcome—death or ICU transfer—within 12 hours, both predicted at triage time, and ED reattendance within 72 hours, predicted at discharge time. As supervised baselines, the study uses logistic regression, random forests, gradient boosting, and multilayer perceptrons fitted on distinct, manually engineered, clinically meaningful tabular features introduced in prior work (68). These features were selected by reviewing prominent ED prediction studies, consulting clinicians and clinical informaticians, and retaining variables that were sufficiently available. MEDS-Tab outperforms all four models trained on these manually designed features on two of the three tasks, despite requiring none of the substantial task-specific feature-engineering effort used to construct that representation. For the two triage-time tasks, the study additionally compares against established, interpretable clinical risk-stratification scores available at or near ED triage, including NEWS (43), NEWS2 (57), REMS (46), MEWS (60), CART (10), and ESI (67). MEDS-Tab outperforms all of these clinical-score baselines on both applicable tasks.

Study 5 (19) compares MEDS-Tab against ICU-XGBoost, a reimplementation of the XGBoost mortality-prediction baseline of Johnson et al. (31). The original study constructs features from selected vital signs, laboratory and blood-gas measurements, Glasgow Coma Scale measurements, and urine output, with the extraction rules detailed in its Section 2.4 and Appendix B, Table 5. Both approaches use XGBoost, but ICU-XGBoost uses a restricted, clinically curated representation, whereas MEDS-Tab constructs sparse features across the dataset's code vocabulary, aggregation functions, and lookback windows. The evaluation includes ten clinically defined and ten randomly generated prediction tasks on each of MIMIC-IV and NWICU, giving 40 dataset–task settings. Using MEDS-TAB-large throughout, MEDS-Tab achieves a strictly higher reported AUROC than ICU-XGBoost in 38/40 settings (95.0%), as calculated from Tables 9–12 of Study 5.

We exclude Study 4 from this baseline comparison because its only comparator is GenHPF, a neural model rather than a conventional tabular baseline.

Across these four studies, MEDS-Tab is evaluated against conventional tabular baselines in 61 study–dataset–task settings and achieves a strictly higher reported AUROC than every such baseline in 55/61 settings (90.2%). These comparisons show that MEDS-Tab's automatically constructed features support strong baselines relative to both simple code-count representations and clinically selected feature sets.

Study 2 Prediction Tasks:

1. Long length of stay (>7 days).
2. In-hospital mortality.
3. 30-day readmission following hospital discharge.
4. Acute myocardial infarction during ED or inpatient visit.
5. Chronic lymphocytic leukemia diagnosis.
6. Celiac disease diagnosis.
7. Hypertension diagnosis.
8. Ischemic stroke during inpatient or ED visit.
9. Metabolic dysfunction-associated steatotic liver disease diagnosis.
10. Osteoporosis diagnosis.
11. Pancreatic cancer diagnosis.
12. Systemic lupus erythematosus diagnosis.
13. Schizophrenia diagnosis.
14. Type 2 diabetes mellitus diagnosis.

E Detailed External Benchmark Aggregation

This appendix section contains copies of externally reported results and describes how win rates and model counts were derived from them to create Table 2. Each comparison represents a study–dataset–task combination for which a study provided AUROC forecasting performance for MEDS-Tab as well as other models. The five studies in total cover 83 comparisons.

E.1 Strict wins, model groups, and pooling

We compute win rates for two groups of models. **All Models** contains all non-MEDS-Tab predictors in the selected comparison's evaluations. **Baselines** is the subset of conventional baselines, listed in Table 4.

For paper p, dataset d, task t, and nonempty model group G_{pdt}, let a_{pdt} be the selected MEDS-Tab AUROC and a_{pdtm} predictor m's AUROC. A task is a strict win only when MEDS-Tab exceeds the best other predictor:

$$w_{pdt,G} = \mathbf{1}\left[a_{pdt} > \max_{m \in G_{pdt}} a_{pdtm}\right].$$

Equality at the published precision does not count as a win. Only reported AUROC point estimates enter this calculation; uncertainty intervals and source Mean/Avg entries are reproduced for completeness, not used as additional tasks or as significance tests.

Each study's percentage is its number of strict wins divided by its number of eligible dataset–tasks. The **All** row pools these counts before division:

$$\text{All Models:} \quad \frac{3+2+0+18+29}{4+14+3+22+40} = \frac{52}{83} \approx 62.7\%,$$

$$\text{Baselines:} \quad \frac{4+11+2+38}{4+14+3+40} = \frac{55}{61} \approx 90.2\%.$$

The # Comparisons column counts all study–dataset–task settings. Study 4's 22 units contribute to All Models but not to Baselines, because it has no comparator in that restricted group. These pooled percentages are descriptive summaries of the selected evaluations, not an assumption that the tasks or studies are statistically independent.

E.2 COUNTING MODELS

The **# Models** column counts distinct models within each study, taking the union of those evaluated across its selected datasets and tasks. It excludes MEDS-Tab, alternative MEDS-Tab configurations, and aggregate entries such as "Best Baseline." A method used for several tasks or datasets is counted once in that study. Separately reported model configurations, such as MOTOR-FT and MOTOR-LP (59), are counted separately; repeated runs and training seeds are not additional methods. Clinical risk scores are included too.

The **# Neural Models** column counts members of All Models that use a neural network for prediction or representation learning, including a linear probe on a neural representation. The **# Baselines** column counts the conventional tabular comparators in Table 4. These subsets overlap: the MLP in Study 3 is both a neural model and a conventional tabular baseline. Clinical scores contribute to # Models, but not to # Neural Models or # Baselines.

The model-count entries in the **All** row sum the within-study counts: 37 study–model entries, including 23 neural and 9 baseline entries. A method appearing in two studies contributes once to each study, so these totals do not represent globally unique methods. Counts describe the methods represented across a study, not necessarily the number available for every task. In Study 3, for example, 13 models appear across the three tasks, but only 12 are evaluated on each triage task and 7 on ED re-presentation. The strict-win rule always uses the methods reported for the particular dataset–task, not the full study-wide union.

Table 4: Explicit membership of the Baselines comparison group. All Models also includes the other eligible predictors named in each study subsection.

External Study	Baselines
Study 1	Logistic regression (Log. Reg.); LightGBM
Study 2	FEMR-GBM (LightGBM); FEMR-LR (logistic regression)
Study 3	LR (logistic regression); RF (random forest); GB (gradient boosting); MLP (multilayer perceptron)
Study 4	None in the selected comparison
Study 5	ICU-XGBoost

Source tables and versions. The thirteen source tables below retain all numerical rows and columns, reported precision, uncertainties, and summary entries. The reported source experiment result values are kept separate from the strict-win counts calculated here.

E.3 STUDY 1: SOURCE TABLE 1

We copy Table 1, "Proof of viability results (AUC-ROC)", from Study 1 (13). Each cell reports MIMIC-IV/Columbia, so two tasks on two datasets give four units. The selected predictor is MEDS-Tab. Baselines contains Log. Reg. and LightGBM (32); All Models additionally includes the neural models MOTOR (59), CEHR-BERT (47), and GenHPF (24), giving 5 comparator models, of which 3 are neural and 2 are Baselines. The source describes these results as preliminary and uses a 10K-patient Columbia subset, distinct from the full Study 2 Columbia cohort. MEDS-Tab strictly wins 3/4 All Models comparisons and 4/4 Baselines comparisons.

Table 5: **Study 1: source Table 1.** Proof of viability results (AUC-ROC). Data reproduced from (13). Each cell retains the source order MIMIC-IV/Columbia.

Model	**MIMIC-IV/Columbia**	
	Long LOS	ICU Mortality
Log. Reg.	0.752/0.677	0.754/0.509
LightGBM	0.783/0.757	0.798/0.661
MOTOR	0.804/0.735	0.854/0.727
CEHR-BERT	0.808/0.741	0.845/0.726
MEDS-Tab	0.811/0.761	0.830/0.785
GenHPF	0.779/0.662	0.790/0.633

E.4 Study 2: source Tables 13, 14, 23 and 24

We use source Tables 13, 14, 23, and 24 from (15). Tables 13–14 report outcome and phenotype discrimination at the largest training-set size; Tables 23–24 report individual baseline results. The selected predictor is MEDS-TAB-XGB from Tables 23–24 on every task. **The "Best Baseline" column is not substituted for MEDS-Tab:** for CLL, for example, it is 0.722, whereas MEDS-TAB-XGB is 0.666. This aggregate column is preserved below but is not an additional comparator.

Baselines contains FEMR-GBM (15) and FEMR-LR (15). All Models additionally includes the seven neural models CEHR-BERT (47), CEHR-GPT (48), CORE-BEHRT (45), LLAMA (64), MAMBA (21), MAMBA-TRANSPORT, and MOTOR (59), giving 9 comparator models and 2 Baselines. MEDS-TAB-LR is reproduced but excluded as a self-comparison. The 14 CUMC tasks give 2/14 strict All Models wins and 11/14 strict Baselines wins. Other training sizes, subpopulation evaluations, and modality ablations are not counted as additional tasks.

Table 6: **Study 2: source Table 13.** Outcome discriminative performance results. Data reproduced from (15). Parenthesized values are preserved exactly as reported; point estimates determine wins.

	Best Baseline	Cehr Bert	Cehr Gpt	Core Behrt	Llama	Mamba	Mamba-Transport	Motor
Long LOS	0.838 (0.002)	0.789 (0.001)	0.864 (0.001)	0.772 (0.002)	0.760 (0.002)	0.787 (0.002)	0.756 (0.002)	0.846 (0.002)
Mortality	0.911 (0.004)	0.868 (0.004)	0.936 (0.003)	0.864 (0.004)	0.847 (0.005)	0.883 (0.004)	0.852 (0.004)	0.958 (0.002)
Readmission	0.765 (0.003)	0.724 (0.003)	0.773 (0.003)	0.727 (0.003)	0.746 (0.003)	0.752 (0.003)	0.717 (0.003)	0.777 (0.003)

Table 7: **Study 2: source Table 14.** Phenotype discriminative performance results. Data reproduced from (15). Parenthesized values are preserved exactly as reported; point estimates determine wins.

Phenotype	Best Baseline	Cehr Bert	Cehr Gpt	Core Behrt	Llama	Mamba	Mamba-Transport	Motor
AMI	0.859 (0.008)	0.807 (0.008)	0.853 (0.007)	0.805 (0.010)	0.806 (0.008)	0.813 (0.009)	0.782 (0.009)	0.845 (0.007)
CLL	0.722 (0.027)	0.668 (0.025)	0.699 (0.031)	0.628 (0.025)	0.602 (0.032)	0.687 (0.032)	0.524 (0.034)	0.584 (0.027)
Celiac	0.570 (0.043)	0.671 (0.036)	0.696 (0.039)	0.650 (0.034)	0.541 (0.039)	0.635 (0.045)	0.628 (0.038)	0.724 (0.038)
HTN	0.728 (0.004)	0.725 (0.004)	0.750 (0.003)	0.718 (0.003)	0.696 (0.004)	0.722 (0.004)	0.689 (0.004)	0.776 (0.003)
Ischemic Stroke	0.894 (0.007)	0.841 (0.009)	0.859 (0.007)	0.844 (0.007)	0.829 (0.008)	0.863 (0.006)	0.811 (0.010)	0.873 (0.006)
MASLD	0.684 (0.008)	0.687 (0.009)	0.728 (0.008)	0.698 (0.008)	0.683 (0.009)	0.709 (0.008)	0.673 (0.009)	0.706 (0.008)
Osteoporosis	0.720 (0.006)	0.708 (0.006)	0.749 (0.006)	0.714 (0.006)	0.663 (0.007)	0.708 (0.006)	0.645 (0.006)	0.746 (0.005)
Pancreatic Cancer	0.779 (0.021)	0.737 (0.018)	0.721 (0.020)	0.782 (0.014)	0.657 (0.024)	0.751 (0.017)	0.669 (0.022)	0.709 (0.024)
SLE	0.697 (0.027)	0.743 (0.032)	0.817 (0.030)	0.769 (0.031)	0.683 (0.031)	0.776 (0.034)	0.649 (0.035)	0.709 (0.030)
Schizophrenia	0.792 (0.021)	0.743 (0.020)	0.812 (0.017)	0.742 (0.017)	0.756 (0.016)	0.751 (0.017)	0.712 (0.022)	0.877 (0.012)
T2DM	0.715 (0.009)	0.748 (0.008)	0.783 (0.008)	0.754 (0.008)	0.698 (0.008)	0.767 (0.007)	0.714 (0.008)	0.807 (0.007)

Table 8: **Study 2: source Table 23.** Outcome prediction results for considered baselines. Data reproduced from (15). Parenthesized values are preserved exactly as reported; point estimates determine wins.

Task	Femr-GBM	Femr-LR	Meds-Tab-XGB	Meds-Tab-LR
Long LOS	0.808 (0.002)	0.727 (0.003)	0.838 (0.002)	0.801 (0.002)
Mortality	0.896 (0.004)	0.833 (0.006)	0.911 (0.004)	0.880 (0.006)
Readmission	0.745 (0.003)	0.682 (0.003)	0.765 (0.003)	0.721 (0.003)

Table 9: **Study 2: source Table 24.** Phenotype prediction results for considered baselines. Data reproduced from (15). Parenthesized values are preserved exactly as reported; point estimates determine wins.

Phenotype	Femr-GBM	Femr-LR	Meds-Tab-XGB	Meds-Tab-LR
AMI	0.775 (0.009)	0.669 (0.014)	0.859 (0.008)	0.751 (0.011)
CLL	0.489 (0.035)	0.722 (0.027)	0.666 (0.025)	0.716 (0.037)
Celiac	0.570 (0.043)	0.529 (0.041)	0.495 (0.035)	0.547 (0.041)
HTN	0.699 (0.004)	0.579 (0.005)	0.728 (0.004)	0.562 (0.005)
Ischemic Stroke	0.850 (0.008)	0.723 (0.013)	0.894 (0.007)	0.710 (0.014)
MASLD	0.668 (0.009)	0.538 (0.009)	0.684 (0.008)	0.564 (0.010)
Osteoporosis	0.673 (0.006)	0.530 (0.009)	0.720 (0.006)	0.581 (0.008)
Pancreatic Cancer	0.671 (0.025)	0.467 (0.027)	0.779 (0.021)	0.533 (0.028)
SLE	0.696 (0.032)	0.399 (0.031)	0.697 (0.027)	0.437 (0.029)
Schizophrenia	0.742 (0.021)	0.745 (0.020)	0.792 (0.021)	0.730 (0.020)
T2DM	0.715 (0.009)	0.577 (0.012)	0.705 (0.008)	0.630 (0.011)

E.5 Study 3: source Tables S3, S4 and S5

We reproduce source Tables S3–S5 from (16). These cover hospitalization at triage, a critical outcome within 12 hours of triage, and ED re-presentation within 72 hours, respectively. All four source metrics and their 95% confidence intervals are retained; only AUROC determines wins.

The selected predictor is MEDS-Tab. Baselines contains LR, RF, GB, and MLP, all trained on manually engineered tabular features. All Models additionally includes Med2Vec (7) and Study 3 on all three tasks; ESI (printed as "esi") (67), NEWS(43), NEWS2(57), REMS(46), MEWS(60), and CART (10) on the first two; and LSTM on the third. MEDS-Tab strictly wins 0/3 All Models comparisons and 2/3 Baselines comparisons, losing the latter on the 12-hour critical-outcome task. Across these tasks, the 13 comparator methods include 4 neural models (MLP, Med2Vec, Study 3, and LSTM (23)) and 4 Baselines (LR, RF, GB, and MLP). The six clinical scores are counted in All Models only. The source label "Study 3 (ours)" refers to the Study 3 authors.

Table 10: **Study 3: source Table S3.** Prediction of Hospitalization At Triage. Data reproduced from (16). All metrics and bracketed 95% confidence intervals are retained; only AUROC determines wins.

	AUROC	AUPRC	Sensitivity	Specificity
LR	0.809 [0.805, 0.813]	0.775 [0.769, 0.781]	0.734 [0.721, 0.750]	0.736 [0.721, 0.749]
Med2Vec	0.816 [0.812, 0.820]	0.782 [0.775, 0.788]	0.751 [0.734, 0.770]	0.728 [0.711, 0.745]
RF	0.817 [0.814, 0.821]	0.785 [0.779, 0.791]	0.759 [0.736, 0.764]	0.726 [0.720, 0.747]
GB	0.819 [0.815, 0.823]	0.792 [0.786, 0.798]	0.753 [0.731, 0.770]	0.728 [0.715, 0.751]
MLP	0.822 [0.818, 0.826]	0.796 [0.790, 0.802]	0.754 [0.743, 0.775]	0.734 [0.716, 0.745]
esi	0.712 [0.707, 0.716]	0.632 [0.625, 0.638]	0.584 [0.577, 0.590]	0.784 [0.779, 0.789]
NEWS	0.581 [0.576, 0.586]	0.555 [0.548, 0.561]	0.563 [0.556, 0.569]	0.546 [0.540, 0.553]
NEWS2	0.565 [0.560, 0.570]	0.538 [0.532, 0.544]	0.519 [0.512, 0.526]	0.570 [0.564, 0.577]
REMS	0.666 [0.661, 0.671]	0.605 [0.598, 0.612]	0.605 [0.552, 0.722]	0.641 [0.545, 0.711]
MEWS	0.558 [0.553, 0.562]	0.521 [0.515, 0.527]	0.296 [0.289, 0.302]	0.812 [0.806, 0.817]
CART	0.673 [0.668, 0.678]	0.617 [0.610, 0.624]	0.703 [0.696, 0.709]	0.578 [0.571, 0.585]
MEDS-Tab	0.863 [0.860, 0.866]	0.879 [0.876, 0.883]	0.746 [0.735, 0.754]	0.820 [0.809, 0.835]
Study 3 (ours)	0.946 [0.944, 0.947]	0.945 [0.943, 0.947]	0.868 [0.859, 0.876]	0.864 [0.856, 0.873]

Table 11: **Study 3: source Table S4.** Prediction of Critical Outcome Within 12h At Triage. Data reproduced from (16). All metrics and bracketed 95% confidence intervals are retained; only AUROC determines wins.

	AUROC	AUPRC	Sensitivity	Specificity
LR	0.875 [0.868, 0.882]	0.308 [0.288, 0.328]	0.813 [0.792, 0.836]	0.782 [0.766, 0.803]
Med2Vec	0.880 [0.872, 0.887]	0.324 [0.305, 0.346]	0.817 [0.799, 0.852]	0.787 [0.762, 0.804]
RF	0.881 [0.873, 0.888]	0.362 [0.343, 0.386]	0.812 [0.794, 0.829]	0.792 [0.788, 0.796]
GB	0.891 [0.884, 0.897]	0.389 [0.367, 0.412]	0.836 [0.804, 0.848]	0.788 [0.779, 0.812]
MLP	0.892 [0.886, 0.898]	0.372 [0.352, 0.396]	0.845 [0.806, 0.855]	0.784 [0.780, 0.823]
esi	0.821 [0.814, 0.829]	0.190 [0.178, 0.201]	0.900 [0.887, 0.913]	0.637 [0.632, 0.642]
NEWS	0.637 [0.624, 0.651]	0.139 [0.127, 0.154]	0.461 [0.440, 0.483]	0.796 [0.792, 0.801]
NEWS2	0.622 [0.610, 0.636]	0.130 [0.118, 0.144]	0.416 [0.402, 0.605]	0.822 [0.533, 0.826]
REMS	0.672 [0.661, 0.683]	0.093 [0.087, 0.102]	0.662 [0.642, 0.683]	0.602 [0.597, 0.607]
MEWS	0.623 [0.611, 0.634]	0.101 [0.093, 0.110]	0.445 [0.424, 0.466]	0.772 [0.768, 0.776]
CART	0.699 [0.687, 0.710]	0.134 [0.123, 0.147]	0.579 [0.557, 0.600]	0.720 [0.716, 0.725]
MEDS-Tab	0.853 [0.846, 0.861]	0.513 [0.493, 0.531]	0.735 [0.717, 0.752]	0.764 [0.759, 0.771]
Study 3 (ours)	0.945 [0.941, 0.950]	0.696 [0.678, 0.712]	0.876 [0.860, 0.898]	0.873 [0.852, 0.889]

Table 12: **Study 3: source Table S5.** Prediction of Emergency Department Re-presentation Within 72h. Data reproduced from (16). All metrics and bracketed 95% confidence intervals are retained; only AUROC determines wins.

	AUROC	AUPRC	Sensitivity	Specificity
LR	0.679 [0.661, 0.697]	0.161 [0.141, 0.185]	0.562 [0.544, 0.645]	0.699 [0.613, 0.718]
Med2Vec	0.621 [0.601, 0.640]	0.128 [0.110, 0.148]	0.560 [0.477, 0.595]	0.615 [0.568, 0.725]
RF	0.673 [0.655, 0.691]	0.150 [0.131, 0.173]	0.642 [0.549, 0.665]	0.599 [0.594, 0.693]
GB	0.697 [0.680, 0.714]	0.165 [0.143, 0.188]	0.623 [0.592, 0.704]	0.662 [0.582, 0.690]
MLP	0.693 [0.675, 0.711]	0.168 [0.147, 0.192]	0.603 [0.579, 0.676]	0.675 [0.607, 0.701]
LSTM	0.689 [0.671, 0.708]	0.164 [0.143, 0.186]	0.595 [0.566, 0.653]	0.680 [0.633, 0.722]
MEDS-Tab	0.714 [0.696, 0.731]	0.189 [0.167, 0.214]	0.645 [0.575, 0.689]	0.657 [0.617, 0.742]
Study 3 (ours)	0.745 [0.728, 0.762]	0.214 [0.190, 0.239]	0.669 [0.611, 0.698]	0.685 [0.657, 0.757]

E.6 STUDY 4: SOURCE TABLE 13

We reproduce source Table 13, "Clinical task utility evaluation", from (17). Both TRTR (train on real, test on real) and TSTR (train on synthetic, test on real) columns are retained. **Only TSTR enters the calculation:** MEDS-TAB is compared with GenHPF on the same dataset and task. This compares downstream predictors/representations, not MEDS-Tab against synthetic-data generators.

GenHPF is the sole All Models model compared against, and is a neural model. Thus this study contributes 1 model, 1 neural model, and 0 Baselines; there is no conventional tabular baseline for the Baselines win rate. The 11 tasks on each of MIMIC-IV and eICU give 22 units: 18 strict wins, two published-precision ties (MIMIC-IV NE and Prop), and two losses (eICU NE and Prop). Both ties are excluded from the win count, giving 18/22 strict wins (81.8%).

Table 13: **Study 4: source Table 13.** Clinical task utility evaluation. Data reproduced from (17). All TRTR/TSTR columns, means, standard deviations, and the Avg row are retained; only TSTR determines wins.

	MIMIC-IV				**eICU**			
	GenHPF		**MEDS-TAB**		**GenHPF**		**MEDS-TAB**	
Task	TRTR	TSTR	TRTR	TSTR	TRTR	TSTR	TRTR	TSTR
Cr	0.89 ± 0.00	0.88 ± 0.00	0.97 ± 0.00	0.95 ± 0.00	0.86 ± 0.00	0.83 ± 0.00	0.94 ± 0.00	0.89 ± 0.00
Plt	0.88 ± 0.01	0.87 ± 0.00	0.96 ± 0.00	0.95 ± 0.00	0.84 ± 0.00	0.83 ± 0.00	0.94 ± 0.00	0.91 ± 0.00
WBC	0.78 ± 0.01	0.73 ± 0.03	0.91 ± 0.00	0.86 ± 0.00	0.73 ± 0.01	0.70 ± 0.02	0.87 ± 0.00	0.80 ± 0.00
HB	0.75 ± 0.01	0.74 ± 0.01	0.85 ± 0.00	0.82 ± 0.00	0.74 ± 0.00	0.72 ± 0.01	0.81 ± 0.00	0.78 ± 0.00
HCO3	0.79 ± 0.01	0.79 ± 0.01	0.90 ± 0.00	0.90 ± 0.00	0.78 ± 0.01	0.76 ± 0.00	0.88 ± 0.00	0.86 ± 0.00
Na	0.80 ± 0.01	0.78 ± 0.01	0.93 ± 0.00	0.92 ± 0.00	0.80 ± 0.01	0.79 ± 0.01	0.91 ± 0.00	0.90 ± 0.00
MgSO4	0.81 ± 0.02	0.81 ± 0.01	0.86 ± 0.00	0.83 ± 0.00	0.73 ± 0.01	0.72 ± 0.01	0.81 ± 0.01	0.75 ± 0.00
Hep	0.68 ± 0.01	0.65 ± 0.00	0.80 ± 0.00	0.74 ± 0.00	0.65 ± 0.00	0.63 ± 0.01	0.72 ± 0.00	0.66 ± 0.00
KCl	0.73 ± 0.01	0.70 ± 0.00	0.80 ± 0.00	0.73 ± 0.00	0.74 ± 0.01	0.66 ± 0.00	0.77 ± 0.00	0.68 ± 0.00
NE	0.96 ± 0.00	0.96 ± 0.00	0.96 ± 0.00	0.96 ± 0.00	0.96 ± 0.00	0.95 ± 0.00	0.95 ± 0.00	0.94 ± 0.00
Prop	0.94 ± 0.00	0.94 ± 0.00	0.95 ± 0.00	0.94 ± 0.00	0.96 ± 0.00	0.95 ± 0.00	0.96 ± 0.00	0.94 ± 0.00
Avg	0.82 ± 0.09	0.80 ± 0.09	0.90 ± 0.06	0.87 ± 0.08	0.80 ± 0.09	0.78 ± 0.10	0.87 ± 0.08	0.83 ± 0.10

E.7 Study 5: source Tables 9–12

We reproduce source Tables 9–12 from (19). Ten generated and ten clinical tasks on each of MIMIC-IV and NWICU give 40 units. MEDS-TAB-large is the selected predictor on every task; we do not choose among MEDS-Tab configurations separately for each task.

Baselines contains ICU-XGBoost. All Models additionally includes the eight neural models MEDS-EIC-AR (41), MOTOR-FT (59), MEDS-EIC-AR-sup (41), MOTOR-LP (59), RETAIN (8), DuETT (35), TECO (53), and BEHRT (38), giving 9 models and 1 Baseline. MOTOR-LP (59) is counted as neural because its linear probe uses a neural representation; it remains a separate reported configuration from MOTOR-FT (59). MEDS-TAB-tiny and MEDS-TAB-LR are reproduced but excluded as self-comparisons. Wins use the displayed AUROCs, not the source's boldface based on unrounded values. The strict All Models wins are 3/10 (MG), 10/10 (MC), 8/10 (NC), and 8/10 (NG), totaling 29/40. ICU-XGBoost exceeds MEDS-TAB-large only on NC8 and NC9, giving 38/40 strict Baselines wins.

Table 14: **Study 5: source Table 9.** Task-level AUROC for MIMIC-IV Generated (MG). Data reproduced from (19). All model rows and the reported Mean column are retained. Mean is not an additional task.

Model	MG1	MG2	MG3	MG4	MG5	MG6	MG7	MG8	MG9	MG10	Mean
MEDS-TAB-large	0.8697	0.9075	0.9344	0.9470	0.9695	0.9779	0.9804	0.9768	0.9782	0.9725	0.9514
MEDS-EIC-AR	0.8500	0.9070	0.9325	0.9357	0.9675	0.9699	0.9624	0.9708	0.9718	0.9700	0.9438
MOTOR-FT	0.8703	0.9107	0.9276	0.9540	0.9712	0.9740	0.9735	0.9783	0.9801	0.9742	0.9514
MEDS-EIC-AR-sup	0.8268	0.8842	0.9051	0.9219	0.9541	0.9594	0.9538	0.9638	0.9630	0.9661	0.9298
MOTOR-LP	0.8700	0.8951	0.8950	0.9499	0.9388	0.9720	0.9707	0.9781	0.9796	0.9746	0.9424
ICU-XGBoost	0.8256	0.7936	0.8079	0.9134	0.9165	0.9596	0.9626	0.9401	0.9584	0.9632	0.9041
RETAIN	0.8361	0.8838	0.9097	0.9125	0.9529	0.9511	0.9328	0.9548	0.9575	0.9637	0.9255
DuETT	0.8259	0.8285	0.8604	0.9155	0.9247	0.9420	0.9581	0.9441	0.9462	0.9557	0.9101
MEDS-TAB-tiny	0.8065	0.7826	0.8182	0.8845	0.9010	0.9268	0.9366	0.9241	0.9357	0.9605	0.8877
TECO	0.7922	0.7682	0.7716	0.8750	0.9071	0.9270	0.9181	0.9109	0.9297	0.9520	0.8752
MEDS-TAB-LR	0.7393	0.8712	0.9023	0.7535	0.8638	0.7815	0.7598	0.8898	0.8974	0.8386	0.8297
BEHRT	0.6343	0.6376	0.5855	0.6295	0.5844	0.5722	0.6115	0.5983	0.6148	0.6223	0.6090

Table 15: **Study 5: source Table 10.** Task-level AUROC for MIMIC-IV Clinical (MC). Data reproduced from (19). All model rows and the reported Mean column are retained. Mean is not an additional task.

Model	MC1	MC2	MC3	MC4	MC5	MC6	MC7	MC8	MC9	MC10	Mean
MEDS-TAB-large	0.8105	0.7847	0.8090	0.8759	0.8187	0.8579	0.8935	0.9006	0.9288	0.9225	0.8602
MEDS-EIC-AR	0.7334	0.7591	0.7991	0.8196	0.8121	0.8363	0.8682	0.8848	0.9063	0.9103	0.8329
MOTOR-FT	0.7625	0.7598	0.8017	0.8376	0.8082	0.7974	0.8481	0.8798	0.9123	0.9095	0.8317
MEDS-EIC-AR-sup	0.6923	0.7476	0.7766	0.8050	0.7980	0.8248	0.8473	0.8466	0.8783	0.9051	0.8121
MOTOR-LP	0.7104	0.7550	0.7365	0.8389	0.7993	0.7952	0.8542	0.8783	0.8070	0.9070	0.8082
ICU-XGBoost	0.7755	0.6551	0.7900	0.8266	0.7283	0.7981	0.8474	0.8817	0.9070	0.8388	0.8049
RETAIN	0.6991	0.7359	0.6992	0.7388	0.7520	0.7491	0.8459	0.7404	0.7513	0.8778	0.7589
DuETT	0.7538	0.7056	0.7918	0.7029	0.7454	0.7571	0.7135	0.8575	0.9075	0.8544	0.7789
MEDS-TAB-tiny	0.6903	0.6878	0.7685	0.7362	0.7253	0.7483	0.7785	0.7697	0.8930	0.8368	0.7634
TECO	0.6054	0.6391	0.6523	0.7100	0.6954	0.7242	0.7762	0.7947	0.6863	0.7846	0.7068
MEDS-TAB-LR	0.6129	0.7179	0.6495	0.6103	0.7647	0.6853	0.6409	0.6457	0.6960	0.8641	0.6887
BEHRT	0.5524	0.7223	0.5664	0.5211	0.6076	0.5771	0.5450	0.4243	0.6220	0.9069	0.6045

F Compute benchmarks

For our compute benchmarks we used the following versions: CaTabRa version 0.1.0 installed on python version 3.9, tsfresh version 0.20.2 installed on Python 3.12, and meds-tab version 0.0.7 installed on Python 3.12.

Table 16: **Study 5: source Table 11.** Task-level AUROC for NWICU Clinical (NC). Data reproduced from (19). All model rows and the reported Mean column are retained. Mean is not an additional task.

Model	NC1	NC2	NC3	NC4	NC5	NC6	NC7	NC8	NC9	NC10	Mean
MEDS-TAB-large	0.7847	0.7700	0.8156	0.8012	0.7957	0.8300	0.8128	0.8437	0.8983	0.8279	0.8180
MEDS-EIC-AR	0.7534	0.7425	0.7826	0.7527	0.7946	0.8237	0.8105	0.8046	0.8898	0.8146	0.7969
MOTOR-FT	0.6186	0.7394	0.6523	0.7737	0.7388	0.7524	0.7456	0.7610	0.6921	0.7548	0.7229
MEDS-EIC-AR-sup	0.6728	0.7288	0.7127	0.7356	0.7577	0.7683	0.7569	0.7486	0.7300	0.7973	0.7408
MOTOR-LP	0.6317	0.7052	0.6258	0.7578	0.7656	0.7647	0.7191	0.7664	0.6907	0.7434	0.7170
ICU-XGBoost	0.7706	0.6163	0.8131	0.6902	0.7186	0.7267	0.7150	0.8589	0.9016	0.7775	0.7589
RETAIN	0.6461	0.6797	0.6791	0.7540	0.6892	0.7212	0.7678	0.7512	0.7017	0.7630	0.7153
DuETT	0.7177	0.6193	0.7490	0.6888	0.7192	0.7457	0.6890	0.7901	0.8542	0.7662	0.7339
MEDS-TAB-tiny	0.6782	0.6396	0.7422	0.7006	0.7179	0.7329	0.6972	0.7676	0.7661	0.7459	0.7188
TECO	0.6059	0.5998	0.6051	0.6617	0.6677	0.6927	0.6892	0.7838	0.6190	0.6820	0.6607
MEDS-TAB-LR	0.5634	0.6195	0.6329	0.6138	0.5342	0.5219	0.6878	0.5518	0.6254	0.5962	0.5947
BEHRT	0.5951	0.6003	0.5738	0.4902	0.5341	0.4828	0.5593	0.5035	0.5856	0.7097	0.5634

Table 17: **Study 5: source Table 12.** Task-level AUROC for NWICU Generated (NG). Data reproduced from (19). All model rows and the reported Mean column are retained. Mean is not an additional task.

Model	NG1	NG2	NG3	NG4	NG5	NG6	NG7	NG8	NG9	NG10	Mean
MEDS-TAB-large	0.7838	0.8020	0.7879	0.7966	0.8589	0.8522	0.8764	0.9131	0.8770	0.9021	0.8450
MEDS-EIC-AR	0.7243	0.7591	0.7768	0.8049	0.8429	0.8349	0.8813	0.9036	0.8549	0.8597	0.8242
MOTOR-FT	0.7707	0.7295	0.7430	0.7832	0.7741	0.7722	0.8524	0.8627	0.8497	0.8460	0.7983
MEDS-EIC-AR-sup	0.6885	0.7289	0.6913	0.7504	0.7776	0.7952	0.8039	0.8272	0.8262	0.8468	0.7736
MOTOR-LP	0.7602	0.6900	0.7264	0.7808	0.7806	0.7476	0.8365	0.8430	0.8441	0.8114	0.7821
ICU-XGBoost	0.6570	0.6521	0.7446	0.7239	0.7047	0.7948	0.7641	0.7696	0.8443	0.8352	0.7490
RETAIN	0.7146	0.7806	0.7371	0.7897	0.7848	0.8219	0.8453	0.8798	0.8367	0.8663	0.8057
DuETT	0.6602	0.7457	0.6989	0.7481	0.7391	0.8040	0.8078	0.8166	0.8426	0.8585	0.7722
MEDS-TAB-tiny	0.6505	0.7239	0.6599	0.7120	0.7379	0.7737	0.7115	0.7234	0.7723	0.7965	0.7262
TECO	0.6679	0.5897	0.6414	0.6414	0.6818	0.6804	0.7609	0.7676	0.7572	0.8684	0.7057
MEDS-TAB-LR	0.6291	0.6673	0.6897	0.6594	0.6726	0.7234	0.7488	0.7704	0.7820	0.7635	0.7106
BEHRT	0.5465	0.5720	0.5488	0.5606	0.5498	0.5605	0.6103	0.6090	0.6187	0.5686	0.5745