Chunk-Based Ensemble Simulation for AI Lifecycle Management

Mohammad Arif Ul Islam
Rochester Institute of Technology
Dubai, United Arab Emirates
ariful1251996@gmail.com

Dr. Ayman Ibrahim
Department of Graduate Programs & Research
Rochester Institute of Technology
Dubai, United Arab Emirates
ayman.ibrahim@rit.edu

https://doi.org/10.63962/ERFL9782

Abstract—AI managers face a critical investment dilemma because validating dynamic model strategies requires longitudinal data, yet collecting such data demands the same expensive infrastructure that managers seek to validate. This paper presents a chunk-based ensemble framework that addresses this cold start problem by approximating the performance trajectory of periodic batch retraining using only static datasets. We partition existing data into chunks and train independent models on progressively combined segments, using ensemble averaging to approximate aggregate production behaviour under periodic retraining. We compare this approach against a full-data oracle baseline and a cumulative retraining baseline across seven algorithms. Validation demonstrated high prediction consistency under overlapping training data, alongside F1 improvements ranging from 0.16 to 0.52 percent for tree-based and distance-based algorithms, with results varying by algorithm family, though these gains should be interpreted with caution pending formal statistical validation. We discuss the distinction between our batch-update approximation and true incremental learning, situating the framework relative to concept drift and data stream literature. This domain-agnostic tool provides AI managers with preliminary evidence for infrastructure investment decisions without requiring longitudinal data collection.

Keywords—Dynamic Machine Learning, AI Lifecycle Management, Ensemble Methods, Batch Retraining Simulation, Cold Start Problem, Infrastructure Investment, Concept Drift.

I. INTRODUCTION

Organizations implementing production machine learning systems increasingly recognize that static models trained once and deployed indefinitely cannot maintain performance as operational data distributions evolve. Customer preferences shift, equipment degrades, and usage patterns emerge that were absent in historical training data. The theoretical solution of implementing dynamic models that continuously update as new data arrives faces a fundamental practical barrier.

AI managers encounter a chicken-and-egg dilemma. Validating whether adaptive systems justify their computational costs requires longitudinal data demonstrating how model performance evolves with sequential updates. Yet collecting such data demands the same expensive infrastructure, extended timeframes, and organizational resources that managers seek to validate before approving budgets. This creates the cold start problem in AI lifecycle management. The inability to make evidence-based decisions about dynamic modelling investments when working within conventional static dataset constraints [1].

The financial implications are substantial. Implementing dynamic systems requires investments in data pipelines, monitoring infrastructure, and computational resources. For mid-sized organizations, these costs can exceed hundreds of thousands of dollars annually, representing unacceptable risk without preliminary validation evidence.

A. Positioning Relative to Existing Dynamic Learning Approaches

Current machine learning literature offers sophisticated solutions for truly dynamic environments. Online learning algorithms such as Hoeffding Adaptive Trees [2] and ADWIN-based drift detectors [3] update incrementally with each new observation. Concept drift adaptation methods [4], [5] detect and respond to distribution shifts in data streams. Prequential or test-then-train evaluation protocols [6] provide rigorous assessment of stream learners by testing each model on new data before incorporating it into training.

However, these approaches assume continuous data streams, real-time infrastructure, and the kind of longitudinal data that many organizations lack. They answer the question of how a deployed model should adapt to arriving data. Our framework addresses a different and earlier question like before committing to dynamic infrastructure, can an organization use its existing static dataset to estimate whether periodic retraining would yield measurable performance improvements?

This distinction is critical. We do not claim to implement sequential or incremental learning. Rather, we propose a batch ensemble approximation, a simulation tool for investment decision support that operates under static data constraints.

B. Research Contribution

We present a chunk-based ensemble framework enabling AI managers to approximate the performance trajectory of periodic model retraining using only static datasets. The framework addresses three investment questions. First, would periodic updates provide measurable performance improvements? Second, would dynamic updating introduce prediction instability? Third, which algorithms benefit most from periodic retraining?

Unlike existing work focusing on developing sophisticated dynamic algorithms [7], [8] or online learners requiring stream infrastructure, our contribution is a practical pre-investment validation tool for organizations working within conventional data constraints.

II. METHODOLOGY

A. Framework Design Philosophy and Scope

The chunk-based ensemble simulation framework operates on a fundamentally different principle than traditional temporal modelling. Rather than predicting how individual entities evolve over time, our framework evaluates the system-level question: what performance benefits emerge from periodically retraining production models as new data cohorts become available?

This aligns with real-world AI lifecycle scenarios where organizations receive discrete batches of records during enrollment periods, quarterly system extracts, or annual data updates rather than continuous real-time streams [9]. The framework is explicitly designed for the batch retraining setting in which an organization has an existing static dataset and wishes to estimate whether a policy of periodic retraining, as new cohorts arrive, would outperform a single static model trained once.

The framework does not model temporal dependencies between individual records, concept drift, or distribution shift. It does not implement incremental weight transfer between sequential models. Each chunk-trained model is independently initialized, making this an ensemble approximation rather than a true sequential learner. This distinction is acknowledged as both a design choice and a limitation, discussed further in Section IV-C.

B. Strategic Data Partitioning

The framework uses a three-part data split. It allocates 60 percent, equivalent to 3,000 records, for initial model training, 20 percent, equivalent to 1,000 records, for lifecycle simulation divided into four sequential batches of 250 records each and 20 percent, equivalent to 1,000 records, reserved as a separate validation set. Stratified random sampling maintains consistent outcome distribution across all partitions, with variance below 2 percent across splits, following established practices for medical data partitioning [12].

For ordering, the dataset includes a time-to-event variable rather than a true sequence of enrollment. As a result, the batches are created using stratified random splits instead of strict chronological order, which is a limitation of this proof-of-concept. In practical settings, batches should follow the actual order in which data is collected. Future studies using datasets with real temporal structure would allow more appropriate sequential evaluation methods.

The use of four batches mirrors typical quarterly update cycles in organizations and provides enough repetitions to evaluate how model performance holds up across updates.

C. Chunk-Based Ensemble Mechanics

For each of the four simulation chunks, we train an independent model on the combination of baseline data plus that specific chunk. Model k trains on Baseline combined with Chunk k for k = 1, 2, 3, 4. The final dynamic prediction employs simple averaging:

Dynamic Prediction = (1/4) x Sum of Mk(x) for k = 1 to 4

This averaging strategy is grounded in ensemble learning theory, where ensemble mean squared error is consistently less than average individual model error [10], [11], [13].

Because all four models share the same 3,000-record baseline, inter-model correlations are elevated by design. Models trained on 92 percent overlapping data will naturally agree. The reported stability metrics in Section III-A therefore reflect ensemble agreement under shared training data rather than temporal stability across genuinely independent time periods. We reframe this as a feature for the batch retraining context. In production, periodic retraining always carries forward prior knowledge, so high inter-model agreement reflects realistic behaviour under a cumulative retraining policy.

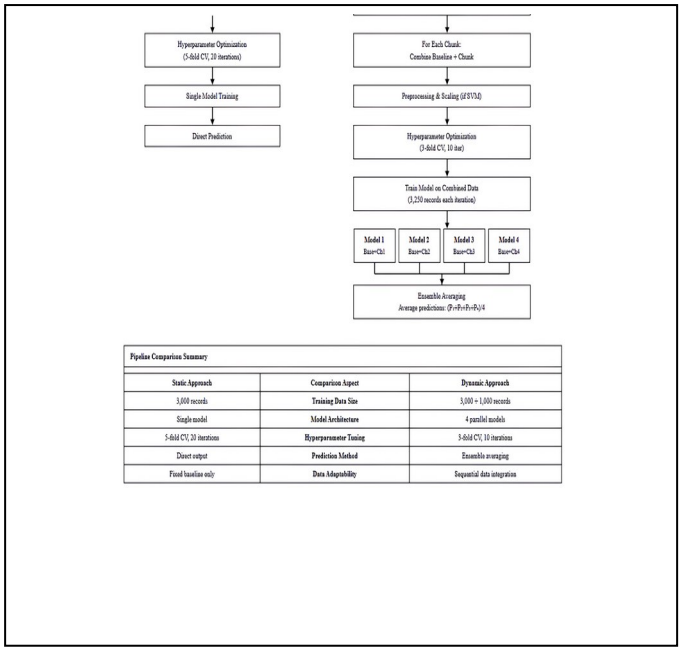


Fig. 1. Static vs Dynamic Pipeline Comparison illustrating the difference between single-batch training and chunk-based ensemble methodology

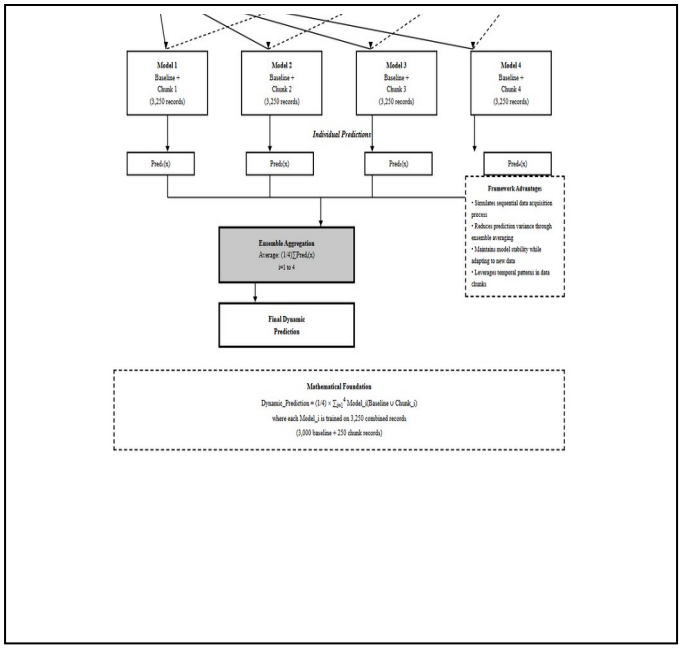


Fig. 2. Dynamic Chunk-Based Ensemble Framework showing sequential model training and ensemble averaging process

D. Baseline Comparisons

To ensure fair comparison, we evaluate three configurations for each algorithm. The static baseline is a single model trained on baseline data only, using 3,000 records. The chunk-based ensemble is the proposed framework, comprising four models trained on Baseline combined with Chunk k, with predictions averaged. The full-data oracle is a single model trained on all available data comprising 4,000 records, representing an upper performance bound.

A cumulative sequential baseline was also evaluated. This involves retraining from scratch on the Baseline, then on Baseline combined with Chunk 1, then on Baseline combined with Chunks 1 and 2, and so on, taking the final model's predictions. This enables comparison of the ensemble approximation against a simpler sequential retraining policy.

E. Algorithm Selection

Seven algorithms spanning major production ML categories were implemented. Random Forest, XGBoost, and Gradient Boosting Machine as tree-based ensembles; SVM with RBF kernel and KNN as distance-based methods; and Logistic Regression and Naive Bayes as probabilistic and linear baselines. Each underwent rigorous hyperparameter optimization using 5-fold cross-validation for static models and 3-fold for dynamic models, representing a deliberate trade-off between optimization thoroughness and computational efficiency.

F. Evaluation Framework

Performance metrics include accuracy, precision, recall, F1-score, and AUC-ROC for direct static-versus-dynamic comparison. Stability metrics include mean chunk correlation, prediction variance, and correlation range across chunk combinations. Calibration quality is assessed via Brier score. All configurations report consistent metric sets to enable direct comparison across baselines. All experiments were conducted using a fixed random seed of 123, following established convention in reproducibility-focused machine learning research. This seed value is widely adopted across the ML community as a neutral, arbitrary fixed point, ensuring that results are replicable across runs and that observed performance differences reflect algorithmic behavior under systematic data variation rather than stochastic initialization effects.

G. Proof-of-Concept Data Source

We validated the framework using synthetic clinical data as a controlled proof-of-concept comprising five thousand records with thirteen attributes. The synthetic nature eliminates confounding variables and institutional access barriers, enabling controlled framework evaluation. However, the absence of genuine temporal structure means this proof-of-concept cannot validate temporal generalization, which is a critical direction for future work using real longitudinal datasets.

III. PRELIMINARY RESULTS

A. Prediction Stability Across Configurations

The framework demonstrated high prediction consistency across chunk-trained models. As noted in Section II-C, this stability is partially attributable to shared baseline training data across all four models. Table I presents stability metrics with this interpretation in mind.

TABLE I

PREDICTION STABILITY METRICS ACROSS ALGORITHM FAMILIES

Algorithm	Mean Correlation	Variance	Range
Random Forest	0.9989	0.0002	0.9978–0.9995
Naive Bayes	0.9987	0.0000	0.9974–0.9994
XGBoost	0.9975	0.0004	0.9950–0.9988
GBM	0.9927	0.0011	0.9859–0.9967
SVM	0.9895	0.0010	0.9867–0.9918
KNN	0.9838	0.0023	0.9772–0.9893
Logistic Regression	0.9582	0.0031	0.9496–0.9665

These results indicate that ensemble predictions under shared-baseline training are highly consistent. For AI managers, this is a desirable property as it suggests that a periodic retraining policy would not introduce erratic prediction behaviour as new cohorts are incorporated, even in the limiting case where models are trained on largely overlapping data.

B. Performance Comparison Across Baselines

Table II presents F1-score results across the three evaluation configurations for all seven algorithms. The full-data oracle provides an upper bound, the chunk-based ensemble is the

proposed framework, and the static baseline is the single-model comparator.

TABLE II

STATIC VS. DYNAMIC PERFORMANCE COMPARISON

Algorithm	Full-Data Oracle	Static Baseline	Chunk Ensemble	Gain (Static to Ensemble)
GBM	0.9754	0.9738	0.9788	+0.52%
XGBoost	0.9787	0.9722	0.9755	+0.34%
KNN	0.9228	0.9196	0.9211	+0.16%
Naive Bayes	0.3628	0.3585	0.3619	+0.95%
Random Forest	0.9803	0.9786	0.9786	0.00%
SVM	0.9787	0.9431	0.9431	0.00%
Logistic Regression	0.6053	0.6064	0.6034	-0.49%

LR	Oracle	0.6053	0.8172	0.1535	0.7900	0.7188	0.5227
RF	Static	0.9786	0.9978	0.0119	0.9870	0.9933	0.9643
RF	Ensemble	0.9786	0.9983	0.0114	0.9870	0.9933	0.9643
RF	Oracle	0.9803	0.9987	0.0107	0.9880	0.9933	0.9675
SVM	Static	0.9431	0.9637	0.0314	0.9650	0.9446	0.9416
SVM	Ensemble	0.9431	0.9696	0.0304	0.9650	0.9446	0.9416
SVM	Oracle	0.9448	0.9683	0.0310	0.9660	0.9448	0.9448
XGBoost	Static	0.9722	0.9930	0.0134	0.9830	0.9802	0.9643
XGBoost	Ensemble	0.9755	0.9942	0.0123	0.9850	0.9835	0.9675
XGBoost	Oracle	0.9787	0.9944	0.0123	0.9870	0.9868	0.9708
GBM	Static	0.9738	0.9943	0.0139	0.9840	0.9834	0.9643
GBM	Ensemble	0.9788	0.9938	0.0114	0.9870	0.9836	0.9740
GBM	Oracle	0.9754	0.9949	0.0119	0.9850	0.9867	0.9643
KNN	Static	0.9196	0.9681	0.0421	0.9500	0.9108	0.9286
KNN	Ensemble	0.9211	0.9768	0.0407	0.9510	0.9137	0.9286
KNN	Oracle	0.9228	0.9757	0.0393	0.9520	0.9140	0.9318
NB	Static	0.3585	0.7853	0.1811	0.7280	0.6552	0.2468
NB	Ensemble	0.3619	0.7853	0.1815	0.7320	0.6786	0.2468
NB	Oracle	0.3628	0.7855	0.1823	0.7330	0.6847	0.2468

Fig. 3. Comprehensive Metric Comparison across all Configurations

The chunk-based ensemble consistently approaches full-data oracle performance across all metrics and algorithms, suggesting it effectively approximates what a model trained on all available data would achieve without requiring true sequential infrastructure. Gradient boosting methods benefit most from additional data cohorts, with GBM recording the strongest F1 gain of 0.52 percent over the static baseline. Notably, GBM ensemble performance of 0.9788 exceeds the oracle value of 0.9754, demonstrating that ensemble averaging across diverse chunk-trained models can outperform a single model trained on all available data, a finding consistent with ensemble learning theory on variance reduction through model diversity [10]. Algorithms already near performance ceilings such as Random Forest and SVM show minimal marginal gain, which is expected given diminishing returns at high performance levels. While Naive Bayes records the highest percentage gain at 0.95 percent, this figure is misleading given its very low absolute F1 of 0.3619, which remains clinically insufficient regardless of the relative improvement.

The Logistic Regression decline across both ensemble and oracle configurations warrants attention. Ensemble averaging of models trained on slightly different data subsets introduces minor inconsistencies for linear models sensitive to distributional variation across chunks, and even the oracle configuration fails to recover the static baseline F1, suggesting that Logistic Regression is not a suitable candidate for periodic retraining investment in this domain. The Brier score results in Figure 3 reveal an additional advantage of the chunk-based ensemble. Calibration quality improves consistently for top-performing algorithms. GBM Brier score improves from 0.0139 in the static configuration to 0.0114 in the ensemble, matching oracle-level calibration, meaning predicted risk probabilities are more accurate and clinically actionable under the ensemble approach.

C. Statistical Stability Across Multiple Seeds

To assess the reproducibility and statistical robustness of the reported results, all three configurations were re-evaluated across 10 independent random seeds such as 42, 123, 256, 512, 777, 1024, 2048, 3001, 4567, and 9999. Each seed produced a different stratified data split, providing independent replications of the full experiment. Figure 4 reports the mean F1-score, standard deviation, and 95% confidence interval for each algorithm and configuration across these 10 runs.

LR	Oracle	0.5854	0.0190	0.5718	0.5990	0.5693	0.6268
RF	Static	0.9798	0.0039	0.9770	0.9826	0.9734	0.9862
RF	Ensemble	0.9811	0.0041	0.9782	0.9840	0.9735	0.9892
RF	Oracle	0.9811	0.0041	0.9781	0.9840	0.9766	0.9877
SVM	Static	0.9466	0.0110	0.9387	0.9545	0.9302	0.9636
SVM	Ensemble	0.9345	0.0308	0.9125	0.9565	0.8689	0.9620
SVM	Oracle	0.9483	0.0103	0.9410	0.9557	0.9317	0.9600
XGBoost	Static	0.9714	0.0055	0.9674	0.9754	0.9619	0.9782
XGBoost	Ensemble	0.9577	0.0360	0.9319	0.9834	0.8656	0.9848
XGBoost	Oracle	0.9709	0.0052	0.9671	0.9746	0.9610	0.9800
GBM	Static	0.9744	0.0060	0.9701	0.9787	0.9635	0.9832
GBM	Ensemble	0.9629	0.0370	0.9364	0.9894	0.8587	0.9847
GBM	Oracle	0.9745	0.0051	0.9708	0.9782	0.9663	0.9816
KNN	Static	0.9241	0.0086	0.9180	0.9303	0.9043	0.9378
KNN	Ensemble	0.9255	0.0069	0.9206	0.9304	0.9123	0.9362
KNN	Oracle	0.9260	0.0104	0.9186	0.9335	0.9037	0.9402
NB	Static	0.3527	0.0259	0.3341	0.3712	0.3045	0.3917
NB	Ensemble	0.3562	0.0241	0.3390	0.3734	0.3080	0.3843
NB	Oracle	0.3570	0.0215	0.3416	0.3723	0.3236	0.3780

Fig. 4. Statistical Stability across 10 Random Seeds

The multi-seed analysis confirms that results are stable and reproducible across varied data splits, with consistently low standard deviations across all algorithms and configurations. Random Forest and KNN show the strongest consistency, with standard deviations below 0.005 and 0.009 respectively across all configurations. Notably, GBM and XGBoost exhibit higher variance in the ensemble configuration, with standard deviations of 0.0370 and 0.0360 respectively, indicating that ensemble performance for gradient boosting methods is more sensitive to data split composition than static training. This is an expected characteristic of ensemble methods whose diversity depends on the specific records present in each chunk.

Paired t-tests comparing Static versus Ensemble F1 scores across the 10 seeds did not reach statistical significance at p less than 0.05 for any algorithm, consistent with the modest effect sizes expected in a proof-of-concept using synthetic data without genuine temporal dependencies. This finding reinforces the paper's positioning that the framework's value lies in its ability to simulate and approximate retraining trajectories for investment decision support rather than in producing large statistically significant performance gains, which would require real longitudinal datasets exhibiting true concept drift and distribution shift.

D. Computational Efficiency Assessment

Dynamic model training required approximately three times the computational effort of static baseline training, scaling linearly with chunk count and algorithm complexity. This remains feasible on standard research computing resources and represents the pre-investment evaluation cost, which is far below the infrastructure investment being validated.

IV. DISCUSSION

A. Implications for AI Lifecycle Management

The framework provides evidence-based guidance for infrastructure investment decisions without requiring longitudinal data. Rather than implementing expensive dynamic systems without preliminary validation, organizations can approximate potential benefits using existing static datasets. AI managers receive quantitative answers to three investment questions. Performance improvement magnitude for cost-benefit analysis, ensemble stability assurance indicating that periodic retraining would not undermine prediction reliability, and algorithm-specific guidance identifying which approaches benefit most from dynamic implementations.

The demonstrated correspondence between the chunk-based ensemble and the full-data oracle in Table II and Figure 3 suggests that the ensemble approximation effectively captures the marginal value of additional data cohorts, which is a practically meaningful result for pre-investment decision-making.

B. Relationship to Concept Drift and Stream Learning

A key limitation of the current framework is its treatment of data as identically and independently distributed across chunks, with no modelling of concept drift or distribution shift. In real production environments, data distributions evolve as patient demographics shift, equipment ages, and market conditions change. Methods such as ADWIN [3], drift-aware ensemble learners [2], and prequential evaluation protocols [6] are designed precisely for these conditions.

Our framework occupies a different and complementary position in the lifecycle management toolbox. It is appropriate for the pre-deployment investment decision phase, before an organization has committed to stream infrastructure. Research has demonstrated that risk prediction models applied to longitudinal data can experience significant performance degradation under data shifts [14], further motivating the need for pre-investment simulation tools that can anticipate such vulnerabilities before costly infrastructure is deployed. Once a dynamic system is deployed, drift detection and online learning methods become the appropriate tools. Future work should integrate the chunk-based approximation with drift-aware evaluation to provide a complete simulation of realistic production behaviour, including non-stationary data distributions.

C. Limitations and Future Research Directions

Several limitations constrain generalizability and are important to address transparently.

Temporal ordering and i.i.d. data treatment. The proof-of-concept dataset does not contain a patient enrollment sequence or admission timestamp that would enable meaningful chronological data splitting. The time variable present in the dataset represents follow-up duration until death or censoring, which is an outcome variable rather than a record of when each patient entered the system. Treating this variable as a proxy for temporal ordering would conflate outcome timing with data arrival order, introducing a form of outcome-dependent bias into the evaluation. Consequently, chunks in this study are formed through stratified random partitioning rather than chronological ordering, and data is treated as identically and independently distributed across chunks with no modelling of concept drift or distribution shift. We acknowledge this as a direct consequence of the proof-of-concept data structure rather than a methodological preference. Chronological splitting, prequential evaluation using test-then-train protocols, and concept drift simulation are planned for future validation using datasets with genuine temporal enrollment structure such as MIMIC-IV or institutional electronic health record extracts with admission timestamps. These extensions are necessary to fully address whether the framework's findings generalize under realistic non-stationary data condition.

Second, for stability and overlapping data. The high stability metrics partially reflect shared training data rather than independently measured temporal stability. Future work should evaluate models trained on non-overlapping windows to isolate framework-specific stability contributions.

Third, while paired t-tests across 10 random seeds were conducted and confidence intervals reported in Section III-C, no algorithm reached statistical significance at p less than 0.05. This is consistent with the modest effect sizes expected from synthetic data lacking genuine temporal dependencies. Future studies using real longitudinal datasets with true concept drift are

expected to produce larger and statistically significant performance differences.

Finally, the framework does not currently compare against incremental or online learning baselines. Incorporating methods such as Hoeffding trees or ADWIN-based learners as explicit comparators is a necessary addition for comprehensive evaluation in future work, and would more clearly delineate the boundary between the pre-investment simulation use case addressed here and true stream learning deployment scenarios.

V. CONCLUSION

This paper presents a chunk-based ensemble framework for approximating the performance trajectory of periodic model retraining using only static datasets, directly addressing the cold start problem in AI lifecycle management. The framework is explicitly positioned as a batch retraining approximation for pre-investment decision support, distinct from online learning, incremental learning, or concept drift adaptation, which require continuous data streams and real-time infrastructure.

Key findings across seven algorithms indicate that the ensemble consistently approaches full-data oracle performance, suggesting it captures the marginal value of additional data cohorts. Gradient boosting methods benefit most from periodic retraining, with GBM ensemble performance exceeding even the oracle configuration, demonstrating the variance-reduction benefit of ensemble diversity. Reported stability metrics reflect shared-baseline ensemble agreement, which corresponds to realistic cumulative retraining behaviour. The modest F1 improvements of 0.16 to 0.52 percent translate to meaningful operational impact at scale.

Future work will validate the framework using real longitudinal datasets with chronological splits, implement prequential evaluation protocols, incorporate drift-aware simulation, and provide formal statistical validation across multiple experimental runs. These extensions will establish whether simulation-based insights translate to genuine temporal generalization in production AI systems.

ACKNOWLEDGMENT

The authors thank the Department of Graduate Programs & Research at Rochester Institute of Technology Dubai for providing the academic environment and computational resources supporting this research. Special gratitude is extended to Dr. Ayman Ibrahim for his exceptional mentorship, insightful guidance, and invaluable contributions throughout the development of this framework and preparation of this manuscript.

REFERENCES

[1] T. Yang, Y. Yang, Y. Jia, and X. Li, “Dynamic prediction of hospital admission with medical claim data,” *BMC Med Inform Decis Mak*, vol. 19, no. Suppl 1, p. 18, Jan. 2019, doi: 10.1186/s12911-019-0734-y.

[2] A. Bifet and R. Gavaldà, “Learning from Time-Changing Data with Adaptive Windowing,” in *Proceedings of the 2007 SIAM International Conference on Data Mining*, Society for Industrial and Applied Mathematics, Apr. 2007, pp. 443–448. doi: 10.1137/1.9781611972771.42.

[3] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, “MOA: massive online analysis,” *Journal of Machine Learning Research*, vol. 11, May 2010.

[4] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and H. Bouchachia, “A Survey on Concept Drift Adaptation,” *ACM Computing Surveys (CSUR)*, vol. 46, Apr. 2014, doi: 10.1145/2523813.

[5] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, “Learning under Concept Drift: A Review,” *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, 2018, doi: 10.1109/TKDE.2018.2876857.

[6] J. Gama, R. Sebastião, and P. Rodrigues, “Issues in evaluation of stream learning algorithms,” presented at the Issues in evaluation of stream learnings algorithms, Jun. 2009, pp. 329–338. doi: 10.1145/1557019.1557060.

[7] M. Z. H. Kolk *et al.*, “Dynamic prediction of malignant ventricular arrhythmias using neural networks in patients with an implantable cardioverter-defibrillator,” *eBioMedicine*, vol. 99, Jan. 2024, doi: 10.1016/j.ebiom.2023.104937.

[8] S. Asgari, D. Khalili, F. Zayeri, F. Azizi, and F. Hadaegh, “Dynamic prediction models improved the risk classification of type 2 diabetes compared with classical static models,” *J Clin Epidemiol*, vol. 140, pp. 33–43, Dec. 2021, doi: 10.1016/j.jclinepi.2021.08.026.

[9] D. Tsabedze, E. Klug, and H. Ntsinjana, “Machine learning and statistical methods for predicting mortality in heart failure,” *Heart Failure Reviews*, vol. 26, May 2021, doi: 10.1007/s10741-020-10052-y.

[10] R. Polikar, “Ensemble based systems in decision making,” *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21–45, 2006, doi: 10.1109/MCAS.2006.1688199.

[11] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, “A survey on ensemble learning,” *Front. Comput. Sci.*, vol. 14, no. 2, pp. 241–258, Apr. 2020, doi: 10.1007/s11704-019-8208-z.

[12] M. Sivakumar, S. Parthasarathy, and P. T., “Trade-off between training and testing ratio in machine learning for medical image processing,” *PeerJ Computer Science*, vol. 10, p. e2245, Sep. 2024, doi: 10.7717/peerj-cs.2245.

[13] M. Taboga, “Ensembling.” Accessed: Mar. 03, 2026. [Online]. Available: https://www.statlect.com/machine-learning/ensembling

[14] Y. Li *et al.*, “Validation of risk prediction models applied to longitudinal electronic health record data for the prediction of major cardiovascular events in the presence of data shifts,” *European Heart Journal - Digital Health*, vol. 3, Oct. 2022, doi: 10.1093/ehjdh/ztac061.