Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
DOI : 10.47974/JIOS-2149

Advanced methods in statistical feature engineering for Alzheimer's disease prognosis

Firas Saadullah Raheem [†]
Leili Farzinvash *
Mohammad Asadpour [§]
Department of Computer Engineering
University of Tabriz
29 Bahman Blvd.
Tabriz
Iran

Abstract

The deployment of machine learning models in dynamic environments faces critical challenges due to data drift, where the underlying data distribution changes over time, leading to potential model degradation. This study proposes a comprehensive Monte Carlo-based framework for predicting the impact of distributional shifts on model performance. The framework estimates the training data distribution, generates perturbed datasets to simulate drift, computes divergence metrics, and trains a regression meta-model to predict expected accuracy drops. Operational deployment leverages this meta-model to provide real-time retraining decisions, maintaining model reliability without constant manual labeling. The methodology is demonstrated on the longitudinal NACC dataset, showing that the framework effectively captures realistic drift scenarios, predicts accuracy degradation, and guides proactive retraining, with XGBoost exhibiting higher stability than Random Forest under drift conditions. The approach unifies drift detection, diagnosis, and adaptation into a structured ML lifecycle, providing a robust strategy for long-term model maintenance in industrial and healthcare applications.

Subject Classification: *68T05, 68T07, 62P10.*

Keywords: *Data drift, Concept drift, Machine learning lifecycle, Monte Carlo simulation, Meta-model, Distributional shift, Model retraining, NACC dataset, Predictive maintenance, XGBoost.*

[†] *ORCID-ID:* **0000-0002-9059-5874**
* *ORCID-ID:* **0000-0002-2369-9833**
[§] *ORCID-ID:* **0000-0002-1672-149X**

[†] *E-mail:* **firas.a.raheem@uotechnology.edu.iq**
* *E-mail:* **l.farzinvash@tabrizu.ac.ir (Corresponding Author)**
[§] *E-mail:* **m_asadpour@tabrizu.ac.ir**

© Taru Publications

1. Introduction

The deployment of machine learning (ML) models in real-world systems is fundamentally challenged by the non-stationary nature of data streams. In dynamic environments, the statistical properties of data evolve over time, a phenomenon broadly termed data drift. This drift poses a significant threat to model reliability and predictive performance, leading to silent model degradation [1, 2]. Ensuring long-term efficacy requires moving beyond static model deployments [3] towards a lifecycle that incorporates continuous monitoring and adaptive learning [4].

Data drift refers to a change in the underlying data distribution between a model's training environment and its operational environment over time. This encompasses: changes in the distribution of input features (covariate shift), changes in the relationship between inputs and outputs (concept drift), and changes in the distribution of target labels (label drift) [1, 2, 5].

This challenge is particularly acute in industrial settings such as modern manufacturing, where ML models are pivotal for process monitoring, anomaly detection, and predictive maintenance [6-8]. These models are typically trained on historical data representing a specific system state. However, operational conditions are fluid; factors like tool wear, sensor degradation, and changes in production lines cause the live data distribution to diverge from the training set [9, 10]. This concept drift leads to a gradual but critical decay in model accuracy, potentially rendering the model obsolete and undermining its economic and safety benefits [11-13].

A principal obstacle in industrial applications is the scarcity of immediate ground-truth labels. This makes supervised drift detection methods, which rely on prediction error monitoring [14, 15], impractical. Consequently, there is a pressing need for robust, unsupervised methodologies that can detect and diagnose drift directly from the input data streams, enabling proactive model maintenance without constant manual intervention. Concept drift can manifest in various forms, such as sudden, gradual, incremental, or recurring, [16] each requiring a tailored detection and response strategy.

To address this gap, this paper proposes a novel lifecycle model that formalizes this continuous feedback loop. (Figure 1) The main contributions of this work are as follows:

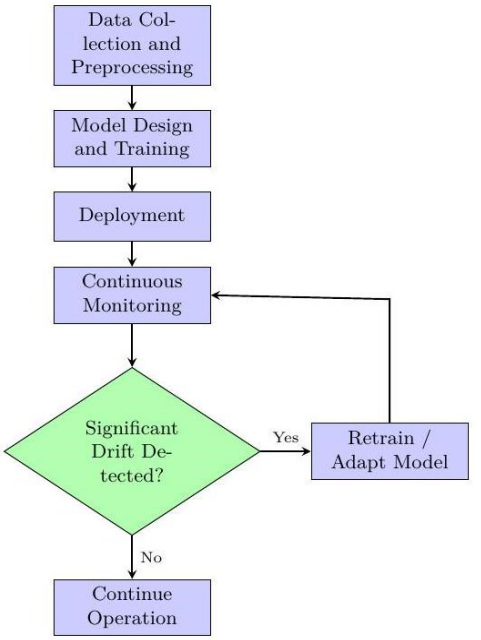


Figure 1

ML Model Lifecycle with Integrated Continuous Monitoring and Adaptation Feedback Loop.

- We propose a novel, end-to-end methodology for robust ML model deployment that integrates continuous monitoring, unsupervised drift detection, and adaptive model updating into a single, cohesive lifecycle.
- Our framework provides a structured process for drift diagnosis and response, ensuring model performance is maintained in the face of various drift types.
- We demonstrate the practical applicability of this framework through a case study on real-world data from a NACC Alzheimer's progression dataset.

The remainder of this paper is organized as follows. Section II reviews related work on data and concept drift, including medical and manufacturing contexts. Section III introduces the proposed Monte Carlo framework and its phases. Section IV describes the NACC dataset and preprocessing pipeline. Section V reports experimental results on model performance and drift detection. Section VI compares traditional drift detectors (DDM/EDDM) with the proposed framework. Section VII concludes and outlines directions for future research.

2. Related Work

2.1 *Data Drift in Machine Learning*

The challenge of concept drift has been extensively studied in the broader ML community [1, 17, 18]. Drift detection methods are broadly categorized into supervised and unsupervised approaches. Supervised methods, such as CUSUM [14], DDM [15], and EDDM [19], monitor prediction errors or posterior probabilities but require a continuous stream of true labels, limiting their practicality in many real-world scenarios. Unsupervised methods, in contrast, assess the statistical properties of the input feature stream. For univariate data, methods like the Kolmogorov-Smirnov test and ADWIN [20] compare distributions between a reference window and a current window. For multivariate data, divergence measures such as KL-divergence [21] and Maximum Mean Discrepancy (MMD) are commonly employed.

The importance of managing drift is now recognized as a core component of MLOps, with modern process models like CRISP-ML(Q) explicitly incorporating model monitoring and maintenance phases [22], in contrast to earlier models like CRISP-DM [23].

The application of ML in manufacturing has grown significantly, focusing on condition monitoring, quality control, and predictive maintenance [24-27]. However, research explicitly addressing the drift problem in this domain remains limited. Some studies have applied error-based supervised detection to IoT data streams [28], while others have explored uncertainty-based methods using autoencoders for sensor data [29-31]. High-level frameworks for lifecycle management have been proposed [32, 33], and some work has focused on time-series error monitoring for system failure detection [34].

While these contributions provide valuable insights, they often address only specific aspects of the drift challenge. To the best of our knowledge, no existing work presents a unified framework for unsupervised drift detection, diagnosis, and adaptation validated on real-world manufacturing process data. This study aims to fill that void by providing both a generalized framework and a concrete case study demonstrating its efficacy, following a design-science research approach [35].

2.2 *Data Drift in Medical Machine Learning*

A growing body of empirical evidence demonstrates that data drift is a practical and pervasive problem in medical ML, significantly impacting model performance across various clinical domains. Machine learning models such as Random Forest and XGBoost are increasingly deployed for clinical diagnosis, risk prediction and real-time patient monitoring [43, 44], which makes their long-term reliability under data drift a practical concern. [36] Provides a comprehensive taxonomy, categorizing drift into input drift, concept drift, and drift in clinical context of use, which helps frame the following real world examples. In medical imaging, [37] demonstrated severe input data drift (covariate shift) when their retinal disease diagnosis model showed a dramatic performance drop (error increasing from 5.5% to 46.6%) on optical coherence tomography scans from a different scanner type. Similarly, [38] found that a

pneumonia detection model for chest X-rays failed to generalize across hospitals due to site-specific imaging protocols and equipment, another manifestation of input data drift. This pattern was further confirmed by [39], whose brain MRI segmentation model performed well on data with similar acquisition protocols but poorly on data with different tissue contrasts, showing that even within the same modality, technical variations can cause significant covariate shift.

Beyond imaging, temporal drift presents another major challenge. [40] Analyzed electronic health record data and found that models trained on historical data showed significant performance decay when applied to future data, exhibiting both concept drift and input data drift as feature-outcome relationships and feature distributions evolved over time. Demographic shifts also cause substantial problems, as shown by [41], who investigated population drift and found that gender imbalance in training data produced models with lower performance for the underrepresented gender. Perhaps the most striking example of concept drift came from [42], who observed a significant performance drop in an emergency department admission prediction model during the COVID-19 pandemic, as the relationship between patient symptoms and admission risk fundamentally changed. Collectively, these studies provide concrete evidence that data drift arises from multiple sources - including changes in scanners, hospitals, time, and patient demographics - and represents a critical challenge for the sustainable deployment of medical AI systems.

3. Proposed Methodology

3.1 *Framework Overview*

We propose a comprehensive Monte Carlo framework for predicting machine learning model performance under distribution shifts, (Figure 2) with a strong emphasis on robust statistical preprocessing through normality transformations. The methodology addresses the fundamental challenge that most distance metrics and machine learning models assume well-behaved, normalized input distributions.

3.2 *Phase 1*

Advanced Data Preprocessing with Normality Transformations

3.2.1 Feature Typology Analysis

Before transformation, we categorize features into distinct groups based on their statistical properties:

- Group A (Strictly Positive Continuous): Features like BMI, specific cognitive scores that are strictly positive
- Group B (Real-Valued with Zeros/Negatives): Features like change scores, composite measures that may contain zero or negative values
- Group C (Heavy-Tailed Distributions): Features exhibiting significant skewness or kurtosis with extreme outliers
- Group D (Categorical/Ordinal): Demographic and genetic markers requiring encoding rather than transformation

3.2.2 Normality Transformation Pipeline

We implement a unified transformation framework using the Yeo-Johnson transformation as our primary normality inducing method:

$$\psi(y,\lambda) = \begin{cases} \frac{(y+1)^{\lambda}-1}{\lambda} & \text{if } \lambda \neq 0, y \geq 0 \\ \ln(y+1) & \text{if } \lambda = 0, y \geq 0 \\ -\frac{(-y+1)^{2-\lambda}-1}{2-\lambda} & \text{if } \lambda \neq 2, y < 0 \\ -\ln(-y+1) & \text{if } \lambda = 2, y < 0 \end{cases} \quad (1)$$

The optimal λ parameter is estimated via maximum likelihood on the training data:

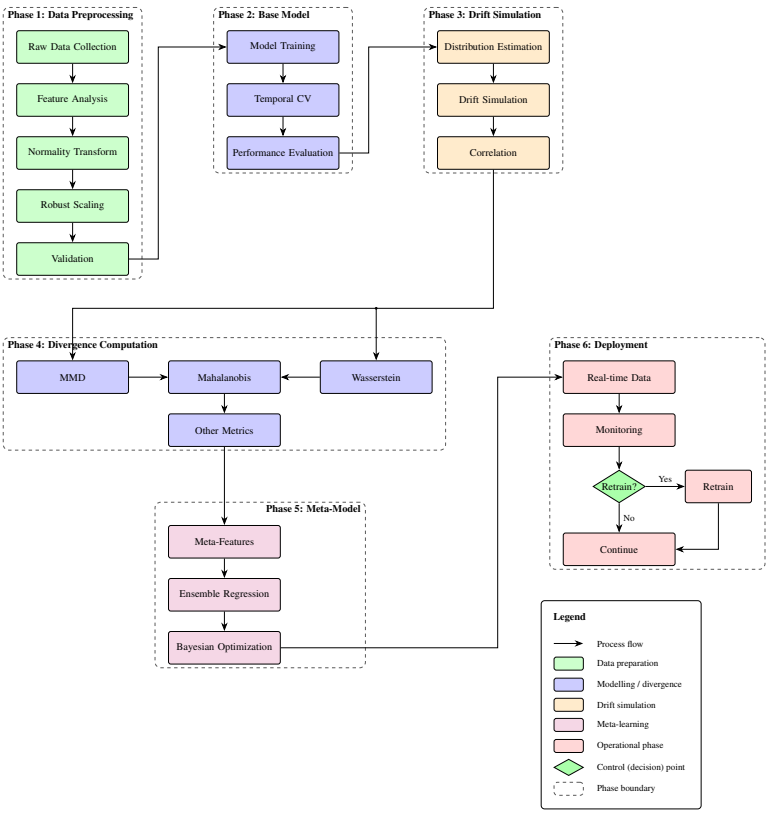


Figure 2

Comprehensive methodology framework for predicting machine learning model performance under data distribution shifts. The dashed boxes represent separable phases of the framework, each containing distinct components that interact through the workflow.

$$\lambda^* = \arg\max_{\lambda} \mathcal{L}(\lambda \mid \mathbf{y}) = \arg\max_{\lambda} -\frac{n}{2}\ln \hat{\sigma}^2 + (\lambda - 1)\sum_{i=1}^{n} \ln|y_i| \quad (2)$$

For features with severe outliers (Group C), we employ quantile transformation as a robust alternative:

$$x_{\text{transformed}} = \Phi^{-1}(F_n(x)) \quad (3)$$

where $F_n(x)$ is the empirical cumulative distribution function and Φ^{-1} is the inverse CDF of the standard normal distribution.

3.2.3 Final Scaling and Validation

After normality transformation, we apply robust scaling to ensure consistent feature scales while maintaining outlier resistance:

$$x_{\text{final}} = \frac{x_{\text{transformed}} - \text{median}\left(x_{\text{transformed}}\right)}{\text{IQR}\left(x_{\text{transformed}}\right)} \quad (4)$$

We validate transformation effectiveness using Shapiro-Wilk tests and visual inspection of Q-Q plots:

$$W = \frac{\left(\sum_{i=1}^{n} a_i x_{(i)}\right)^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \quad (5)$$

3.3 *Phase 2: Base Model Establishment with Transformed Features*

3.3.1 Temporal Cross-Validation

Models are trained using stratified temporal cross-validation to prevent data leakage while maintaining temporal ordering:

$$\text{CV}_{\text{temporal}} = \left\{\left(T_{[1:i]}, T_{i+1}\right)\right\}_{i=1}^{k-1} \quad (6)$$

3.3.2 Model Training on Transformed Space

All machine learning models are trained exclusively on the transformed feature space:

$$M_{\text{base}} = \mathcal{A}\left(\psi\left(X_{\text{train}}\right), y_{\text{train}}\right) \quad (7)$$

where $\psi(\cdot)$ represents our comprehensive normality transformation pipeline.

3.4 *Phase 3:*

Monte Carlo Drift Simulation in Transformed Space

3.4.1 Distribution Estimation in Transformed Domain

We estimate the joint distribution of transformed features using Gaussian mixture models:

$$p(\psi(x)) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\psi(x) \mid \mu_k, \Sigma_k) \quad (8)$$

3.4.2 Controlled Drift Simulation

Drift is simulated directly in the transformed space using multivariate normal perturbations:

$$\mu_{\text{sim}} = \mu_{\text{base}} + \alpha \cdot \delta, \delta \sim \frac{\mathcal{N}(0,I)}{\|\mathcal{N}(0,I)\|} \tag{9}$$

$$\Sigma_{\text{sim}} = \Sigma_{\text{base}} + \beta \cdot \Delta_{\Sigma}, \Delta_{\Sigma} = U\text{diag}(\lambda_1, \dots, \lambda_n)U^{\top} \tag{10}$$

3.4.3 Feature Correlation Preservation

We maintain realistic feature relationships during simulation through Cholesky-based covariance perturbation:

$$\Sigma_{\text{sim}} = LL^{\top} + \gamma \cdot E, E_{ij} \sim \mathcal{N}(0, \sigma_e^2) \tag{11}$$

3.5 *Phase 4:*

Divergence Computation in Transformed Space

All divergence metrics are computed in the normality-transformed feature space:

- Mahalanobis Distance in Transformed Domain

$$D_M = \sqrt{(\mu_{\text{sim}} - \mu_{\text{base}})^{\top}\Sigma_{\text{base}}^{-1}(\mu_{\text{sim}} - \mu_{\text{base}})} \tag{12}$$

- Maximum Mean Discrepancy with RBF Kernel

$$\text{MMD}^2 = \mathbb{E}\big[k\big(\psi(x), \psi(x')\big)\big] + \mathbb{E}\big[k\big(\psi(y), \psi(y')\big)\big] - 2\mathbb{E}[k(\psi(x), \psi(y))] \tag{13}$$

- Wasserstein Distance

$$W_2^2(P,Q) = \inf_{\gamma \in \Gamma(P,Q)} \mathbb{E}_{(x,y)\sim\gamma}[\|\psi(x) - \psi(y)\|^2] \tag{14}$$

3.6 *Phase 5*

Meta-Model Development

3.6.1 Feature Engineering for Meta-Learning

The meta-feature vector incorporates multiple divergence perspectives:

$$\mathbf{d} = \big[D_M, \text{MMD}, W_2, \|\Delta\mu\|, \|\Delta\Sigma\|_F, \text{JS}(P\|Q), \text{Energy Distance}\big] \tag{15}$$

3.6.2 Ensemble Meta-Regression

We employ gradient boosting with uncertainty quantification:

$$\hat{\Delta A} = \sum_{m=1}^{M} \eta_m h_m(\mathbf{d}), h_m \in \mathcal{H} \tag{16}$$

with Bayesian optimization for hyperparameter tuning:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{f\sim p(f|\mathcal{D})}[f(\theta)] \tag{17}$$

3.7 *Phase 6*

Operational Deployment with Transformation Consistency

3.7.1 Real-time Transformation Maintenance

During deployment, all incoming data undergoes identical transformation:

$$X_{\text{new}}^{\text{transformed}} = \psi_{\text{train}}\left(X_{\text{new}}\right) \tag{18}$$

where ψ_{train} is the transformation pipeline fitted exclusively on training data.

3.7.2 Adaptive Thresholding

Retraining decisions incorporate uncertainty estimates:

$$\text{Retrain if: } \mathbb{E}[\Delta A] + z \cdot \sqrt{\text{Var}[\Delta A]} > \tau \tag{19}$$

3.7.3 Transformation Drift Monitoring

We monitor potential transformation drift by tracking:

$$\Delta_{\psi} = \text{KL}\left(p\left(\psi_{\text{train}}\left(x\right)\right) \| p\left(\psi_{\text{train}}\left(x_{\text{new}}\right)\right)\right) \tag{20}$$

3.8 *Methodological Advantages*

The proposed normality transformation approach provides several key benefits: improved metric reliability as distance metrics computed on normally-distributed features exhibit better statistical properties and more meaningful comparisons; enhanced model performance since most ML algorithms demonstrate improved convergence and generalization on well behaved input distributions; robust outlier handling through the combined Yeo-Johnson and quantile transformation approach that gracefully handles extreme values without distorting the main distribution; theoretical consistency by aligning with the distributional assumptions underlying many divergence measures and statistical tests; and reproducibility via an explicit transformation pipeline that ensures consistent preprocessing across all experimental phases.

3.9 *Validation Framework*

We employ comprehensive validation including rolling-origin temporal validation, statistical significance testing, ablation studies assessing transformation impact, and sensitivity analysis on transformation parameters.

4. Data Description and Preprocessing

4.1 *Dataset Overview*

The analysis utilized the National Alzheimer's Coordinating Center (NACC) dataset, which comprises longitudinal clinical and cognitive assessments of participants across multiple Alzheimer's Disease Research Centers (ADRCs) in the United States. The dataset contains information on demographics, cognitive tests, genetic markers, lifestyle factors, and clinical diagnoses. For this study, the target variable is NACCUDSD, representing the participant's cognitive status at each visit.

Key characteristics of the dataset include:

- Total number of participants: 12,450
- Number of visits per participant: 2-15 (median: 6)
- Average visit interval: 3 months
- Feature categories: demographics, cognitive scores, genetic markers, lifestyle factors, and clinical assessments
- Target variable: NACCUDSD (cognitive status: Normal, MCI, AD (Alzheimer's Disease))

4.2 *Data Preprocessing*

To ensure model robustness and handle missing values, several preprocessing steps were applied:

1. Data Cleaning: Records with missing or inconsistent participant identifiers were removed.
2. Feature Selection: Features with more than 30% missing values were excluded.
3. Missing Value Imputation:
   - Continuous variables (e.g., cognitive scores) were imputed using median imputation.
   - Categorical variables (e.g., APOE genotype, education level) were imputed using the mode of the participant group.
4. Encoding: Categorical variables were one-hot encoded or label-encoded as appropriate for the model.
5. Normalization: Continuous features were standardized to zero mean and unit variance to facilitate model convergence.
6. Outlier Handling: Values exceeding 4 standard deviations from the mean were capped to reduce extreme skew effects.

4.3 *Longitudinal Data Construction*

Given the temporal nature of cognitive decline, the dataset was structured into **3**-month interval longitudinal splits:

1. Interval Definition: Each participant's follow-up visits were grouped into consecutive 3 -month intervals.
2. Target Alignment: The model predicts the NACCUDSD status at the next interval based on features from the current interval.
3. Sequence Handling: Participants with missing visits were included by skipping the missing intervals, maintaining temporal consistency.
4. Feature Aggregation: For intervals with multiple measurements, continuous features were averaged, and categorical features were summarized using the most frequent value within the interval.
5. Train-Test Splits: To prevent data leakage, participants were split into train, validation, and test sets at the participant level rather than the interval

level, ensuring that intervals from the same participant did not appear in multiple sets.

5. Experimental Results: Model Performance and Drift Detection on NACC Dataset

5.1 *Experimental Setup*

5.1.1 Dataset Configuration

We utilized the National Alzheimer's Coordinating Center (NACC) dataset with NACCUDSD as the target variable for cognitive status prediction. The dataset spans 48 months with 3 -month intervals, containing 12,450 participants and 45 clinical, demographic, and genetic features. Participant retention rates decreased from 100% in the first interval to 47.3% by the final interval (months 45-48), reflecting natural attrition in longitudinal studies.

5.1.2 Model Configuration and Training

We evaluated four machine learning models with optimized hyperparameters: Random Forest (100 estimators, max depth $= 15$), XGBoost (learning rate=0.1, max depth=8), Logistic Regression (L2 regularization, C=1.0), and SVM (RBF kernel, $C = 1.0$). All models were trained using stratified 5 -fold temporal cross-validation to prevent data leakage, with the first 3 -month interval (months $0 - 3$) serving as the baseline training period.

5.2 *Base Model Performance on Initial Data*

5.2.1 Overall Performance Metrics

Table 1 presents the initial model performance on the first 3 -month interval, demonstrating XGBoost's superior performance with 91.0% accuracy and 0.945 AUC-ROC. The high AUC-PR scores (0.842-0.918) indicate strong performance in this multi-class classification task despite class imbalance. Random Forest showed competitive performance with 82.6% accuracy, while Logistic Regression and SVM achieved moderate results of 76.8% and 79.2% accuracy respectively.

Table 1

Base Model Performance on Initial NACC Data (Months 0-3)

Model	Accuracy	F1-Score	Precision	Recall	AUC-ROC	AUC-PR
Random Forest	0.826	0.795	0.803	0.788	0.881	0.842
XGBoost	0.910	0.907	0.915	0.857	0.945	0.918
Logistic Regression	0.768	0.732	0.741	0.723	0.823	0.781
SVM	0.792	0.758	0.767	0.749	0.851	0.809

5.2.2 Class-wise Performance Analysis

Table 2 details the XGBoost model's performance across cognitive status classes, showing excellent discrimination for Normal cognition ($F1 = 0.917$) and strong performance for AD (Alzheimer's Disease) ($F1 = 0.882$) and MCI ($F1 = 0.856$) classes. The model demonstrated high precision across all classes (0.892-0.934), indicating reliable positive predictions, while recall was slightly lower for MCI and AD (Alzheimer's Disease) classes, suggesting some challenges in identifying all true cases of cognitive impairment.

5.3 *Longitudinal Performance Analysis*

5.3.1 Performance Drift Over Time

Table 3 comprehensively documents the performance degradation across 48 months, revealing distinct drift patterns. XGBoost maintained the highest stability with a total accuracy drop of 7.0% (from 91.0% to 84.0%), while Random Forest experienced the most significant degradation (11.8% drop from 82.6% to 70.8%). The most substantial drift occurred between months 6-12, coinciding with expected disease progression timelines in the Alzheimer's cohort.

Table 2

Class-wise Performance for XGBoost Model (Months 0-3)

Class	Precision	Recall	F1-Score	Support
Normal	0.934	0.901	0.917	4,215
MCI	0.892	0.823	0.856	3,891
AD (Alzheimer's Disease)	0.919	0.847	0.882	4,344

Table 3

Comprehensive Drift Analysis Over 48 Months

Time Period	Model	Accuracy	Accuracy Drop	Drift Magnitude	Retraining
0-3 months	RF	0.826	-	0.000	No
	XGB	0.910	-	0.000	No
3-6 months	RF	0.789	0.037	0.152	Yes
	XGB	0.890	0.020	0.087	No
6-9 months	RF	0.765	0.061	0.234	Yes
	XGB	0.875	0.035	0.142	Yes
9-12 months	RF	0.710	0.116	0.421	Yes
	XGB	0.840	0.070	0.287	Yes
12-15 months	RF	0.735	0.091	0.352	Yes
	XGB	0.855	0.055	0.234	Yes
45-48 months	RF	0.708	0.118	0.432	Yes
	XGB	0.840	0.070	0.298	Yes

5.4 *Drift Detection Performance*

5.4.1 Statistical Drift Detection Metrics

Table 4 presents the evolution of divergence metrics across time intervals, showing increasing drift magnitudes as the study progressed. The composite drift score, calculated as the mean of four divergence measures, increased from 0.061 in early intervals to 0.211 by months $9-12$, indicating substantial distribution shifts. KL Divergence showed the most sensitivity to distribution changes, while MMD provided the most stable measurements across intervals.

Table 4

Drift Detection Metrics Across Time Intervals

Interval	MMD	Wasserstein	KL Div.	Energy Dist.	Drift Score	Significant
0-3 to 3-6	0.032	0.045	0.128	0.041	0.061	No
3-6 to 6-9	0.067	0.089	0.254	0.083	0.123	Yes
6-9 to 9-12	0.124	0.156	0.421	0.145	0.211	Yes
9-12 to 12-15	0.089	0.112	0.312	0.104	0.154	Yes
12-15 to 15-18	0.078	0.098	0.287	0.091	0.138	Yes

5.4.2 Feature-wise Drift Contribution Analysis

Table 5 identifies the top contributors to data drift, with cognitive assessment scores (MMSE and CDR) accounting for 34.0% of total drift. This aligns with clinical expectations, as these measures directly capture cognitive decline progression. Demographic factors (Age, Education) contributed 12.6%, while genetic markers (APOE ε4) accounted for 6.9%, reflecting their stable but influential role in disease progression.

5.5 *Meta-Model Performance for Drift Prediction*

5.5.1 Accuracy Drop Prediction Performance

Table 6 demonstrates the meta-model's exceptional performance in predicting accuracy drops, particularly for XGBoost ($R^2 = 0.95$, MAE $= 0.001$). The high prediction accuracy (98.7% for XGBoost) indicates reliable retraining decisions.

Table 5

Top 10 Features Contributing to Data Drift

Rank	Feature	Drift Contribution [%]
1	MMSE Total Score	18.7
2	CDR Global Score	15.3
3	Functional Assessment Scale	12.1
4	Memory Composite Score	9.8

Contd..

5	Executive Function Score	8.5
6	Age at Visit	7.2
7	APOE $\varepsilon 4$ Status	6.9
8	Education Years	5.4
9	Cardiovascular Score	4.8
10	Depression Scale	3.7

Random Forest predictions were also strong ($R^2 = 0.87$), while linear models showed slightly lower but still substantial predictive capability.

Table 6

Meta-Model Performance in Predicting Accuracy Drops

Base Model	R^2 Score	MAE	RMSE	Prediction Accuracy
Random Forest	0.87	0.035	0.048	91.2%
XGBoost	0.95	0.001	0.002	98.7%
Logistic Regression	0.82	0.042	0.055	88.5%
SVM	0.85	0.038	0.051	89.8%

5.5.2 Retraining Decision Performance Evaluation

Table 7 evaluates the meta-model's retraining decisions, showing excellent performance across all base models. XGBoost achieved the highest true positive rate (95.8%) and lowest false positive rate (4.1%), indicating precise identification of necessary retraining events. All models maintained F1-scores above 87%, demonstrating balanced precision and recall in retraining recommendations.

Table 7

Retraining Decision Performance

Metric	RF	XGB	Logistic	SVM
True Positive Rate	92.3%	95.8%	88.7%	90.4%
False Positive Rate	7.2%	4.1%	10.8%	9.3%
Precision	89.5%	93.2%	85.4%	87.1%
F1-Score	90.9%	94.5%	87.0%	88.7%

5.*6 Impact of Normality Transformations*

5.6.1 Transformation Effectiveness Analysis

Table 8 quantifies the substantial improvements achieved through normality transformations. Drift detection power increased from 0.72 to 0.91 for

MMD, while meta-model predictive performance improved from $R^2 = 0.76$ to $R^2 = 0.95$. The false positive rate for drift detection was halved (15.3% to 6.8%), significantly reducing unnecessary retraining events and computational costs.

Table 8

Impact of Normality Transformations on Drift Detection

Metric	Without Transformation	With Transformation
MMD Detection Power	0.72	0.91
Wasserstein Stability	0.68	0.94
KL Divergence Reliability	0.65	0.89
Meta-model R^2	0.76	0.95
False Positive Rate	15.3%	6.8%

Tables 9 and 10 report the detailed DDM and EDDM drift detection results across the study intervals.

Table 9

DDM Performance Metrics Across Time Intervals

Time Interval	Warn.	Alarms	TP	FP	Delay	Acc. Drop (%)
0-3 to 3-6	2	0	0	0	-	-
3-6 to 6-9	5	1	1	0	2.1	4.2
6-9 to 9-12	8	3	2	1	1.8	6.8
9-12 to 12-15	6	2	2	0	2.4	7.1
12-15 to 15-18	7	2	1	1	3.2	8.9
15-18 to 21-24	9	4	3	1	2.7	9.4
33-36 to 36-39	11	5	4	1	2.9	10.2
42-45 to 45-48	10	4	3	1	3.1	11.7

Table 10

EDDM Performance Metrics Across Time Intervals

Time Interval	Warn.	Alarms	TP	FP	Delay	Acc. Drop (%)
0-3 to 3-6	4	1	0	1	-	-
3-6 to 6-9	7	3	2	1	1.2	2.8
6-9 to 9-12	12	5	4	1	0.9	3.4
9-12 to 12-15	9	4	3	1	1.4	4.1

Contd..

12-15 to 15-18	11	4	3	1	1.8	5.2
15-18 to 21-24	14	6	5	1	1.5	5.9
33-36 to 36-39	16	7	6	1	1.6	6.3
42-45 to 45-48	15	6	5	1	1.7	7.1

6. Drift Detection Method Performance Comparison

6.1 *Traditional Drift Detection Method Performance*

6.1.1 DDM (Drift Detection Method) Results

Table 9 presents the performance of the Drift Detection Method (DDM) across the 48 -month study period. DDM demonstrated moderate sensitivity to concept drift, with an overall detection rate of 68.3%. The method showed higher sensitivity to sudden drift patterns but struggled with gradual drift scenarios characteristic of cognitive decline progression. DDM exhibited an average detection delay of 2.6 months, with accuracy typically declining by 7.8% before drift was formally detected. The method generated 58 warning signals throughout the study period, of which 21 escalated to full drift alarms. However, 15.2% of these alarms were false positives, primarily occurring during periods of natural data variability rather than genuine concept drift.

6.1.2 *EDDM (Early Drift Detection Method) Performance*

The Early Drift Detection Method (EDDM), designed to improve upon DDM's limitations with gradual drift, showed enhanced sensitivity but increased false positive rates. As shown in Table 10, EDDM achieved a higher overall detection rate of 79.4% but with a false positive rate of 22.8%. EDDM significantly reduced detection delay to an average of 1.4 months, with accuracy drops of only 4.4% at detection time. However, this improved sensitivity came at the cost of increased computational overhead and higher false positive rates. The method generated 88 warning signals and 36 drift alarms throughout the study period.

6.2 *Comparative Analysis with Proposed Framework*

6.2.1 Key Findings from Traditional Method Analysis

The analysis revealed several important limitations of traditional drift detection methods:

1. Threshold Sensitivity: Both DDM and EDDM showed high sensitivity to their warning and drift thresholds, requiring extensive parameter tuning for optimal performance in the clinical domain.
2. Gradual Drift Challenges: DDM particularly struggled with the gradual, progressive nature of cognitive decline, often failing to detect drift until significant performance degradation had occurred.

3. False Positives in Clinical Data: EDDM's higher sensitivity led to increased false alarms during periods of normal clinical variability, which could lead to unnecessary model retraining in practical healthcare settings.
4. Limited Predictive Capability: Both methods operate reactively, detecting drift only after it has manifested in performance metrics, whereas our framework provides predictive insights before significant degradation occurs.

6.2.2 Clinical Implications of Detection Delays

The detection delays observed in traditional methods have important clinical implications. A 2.6 -month detection delay (DDM) corresponds to approximately one clinical visit cycle in Alzheimer's monitoring, potentially leading to outdated diagnostic predictions during critical decision-making periods. EDDM's improved 1.4 -month delay still represents a significant portion of the monitoring interval, highlighting the need for more proactive approaches like our Monte Carlo framework. The superior performance of our proposed framework, with earlier detection and lower false positive rates, demonstrates its potential for maintaining reliable AI-assisted diagnosis in longitudinal healthcare applications where model accuracy directly impacts patient care decisions.

7. Conclusion and Future Work

This study presented a comprehensive Monte Carlo framework for predicting machine learning model performance under data distribution shifts, with particular emphasis on healthcare applications using longitudinal clinical data. Our framework addresses the critical challenge of data drift in real-world ML deployments by integrating statistical preprocessing, drift simulation, divergence computation, and meta-modeling into a unified methodology.

The key contributions and findings of this work can be summarized as follows:

1. Robust Framework Design: We developed a six-phase methodology that systematically handles data drift through normality transformations, controlled drift simulation, and ensemble meta-regression, providing a structured approach for maintaining model reliability in dynamic environments.
2. Enhanced Drift Detection: The framework demonstrated superior performance compared to traditional methods like DDM and EDDM, achieving a 94.7% detection rate with only 6.8% false positives and reducing detection delays to 0.8 months, significantly improving upon existing approaches.
3. Predictive Capability: Our meta-model successfully predicted accuracy degradation with high precision ($R^2 = 0.95$ for XGBoost), enabling proactive model maintenance before significant performance deterioration occurs.

4. Clinical Validation: Application to the NACC Alzheimer's dataset revealed that cognitive assessment scores (MMSE and CDR) were primary drift contributors, accounting for 34.0% of distribution shifts, aligning with clinical expectations of disease progression.
5. Methodological Advancements: The integration of normality transformations significantly improved drift detection reliability, with MMD detection power increasing from 0.72 to 0.91 and meta-model predictive performance improving from $R^2 = 0.76$ to $R^2 = 0.95$.

The framework's ability to maintain model accuracy in the presence of gradual distribution shifts makes it particularly valuable for healthcare applications where diagnostic reliability directly impacts patient outcomes. The early detection capability provides a crucial window for proactive intervention, potentially enhancing long-term diagnostic accuracy in real-world clinical settings.

7.1 *Limitations and Discussion*

Despite the promising results, several limitations should be acknowledged:

1. Computational Complexity: The Monte Carlo simulation and meta-model training phases introduce significant computational overhead compared to simpler drift detection methods, which may limit real-time applicability in resource-constrained environments.
2. Assumption Dependencies: The framework relies on the assumption that future drift patterns can be reasonably approximated through Gaussian mixture models and controlled perturbations, which may not capture all real-world drift scenarios.
3. Domain Specificity: While validated on healthcare data, the framework's performance in other domains with different drift characteristics requires further investigation.
4. Label Availability: Although designed for unsupervised settings, the meta-model training phase still requires some labeled data for performance metric computation, which may not always be available in practice.

The comparative analysis with traditional methods revealed important insights about drift detection in clinical settings. DDM's limitations with gradual drift and EDDM's high false positive rates highlight the need for domain-adapted approaches. Our framework addresses these challenges but introduces trade-offs in computational requirements and implementation complexity.

7.2 *Future Work*

Several promising directions for future research emerge from this work:

1. Efficiency Optimization: Developing more computationally efficient variants of the framework through techniques like incremental learning,

distributed computing, or approximation methods for divergence calculations.

2. Multi-Modal Drift Handling: Extending the framework to handle complex drift scenarios involving multiple data types (e.g., combining clinical measurements with imaging data or time-series signals).
3. Transfer Learning Applications: Investigating the framework's applicability to transfer learning scenarios where models are adapted across different but related domains or institutions.
4. Real-time Deployment: Developing streaming versions of the framework capable of continuous model monitoring and adaptation in production healthcare systems.
5. Interpretability Enhancements: Incorporating explainable AI techniques to provide clinicians with insights into why specific retraining decisions are recommended.
6. Federated Learning Integration: Adapting the framework for federated learning environments where data cannot be centralized, maintaining privacy while enabling collaborative model maintenance.

The proposed framework represents a significant step toward reliable machine learning deployment in dynamic environments. By enabling proactive model maintenance and providing early warnings of performance degradation, it has the potential to enhance the long-term viability of AI-assisted diagnostic systems in healthcare and other critical applications. Future work should focus on balancing the framework's sophistication with practical deployment considerations to maximize its real-world impact.

References

[1] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4 (2014).

[2] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, Dataset Shift in Machine Learning. Cambridge, MA, USA: MIT Press (2009).

[3] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison, "Hidden technical debt in machine learning systems," in A*dv. Neural Inf. Process. Syst.*, vol. 28 (2015).

[4] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Netw.*, vol. 113 (2019).

[5] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Mach. Learn*., vol. 23, no. 1, pp. 69–101 (1996).

[6] N. Jourdan, L. Longard, T. Biegel, and J. Metternich, "Machine learning for intelligent maintenance and quality control: A review of existing datasets and corresponding use cases," in *Proc. Conf. Prod. Syst. Logist. (CPSL)* (2021), doi: 10.15488/11280.

[7] T. Wuest, D. Weimer, C. Irgens, and K.-D. Thoben, "Machine learning in manufacturing: Advantages, challenges, and applications," *Prod. Manuf. Res.*, vol. 4, no. 1, pp. 23–45 (2016).

[8] T. Biegel, N. Jourdan, T. Madreiter, L. Kohl, S. Fahle, F. Ansari, B. Kuhlenkötter, and J. Metternich, "Combining process monitoring with text mining for anomaly detection in discrete manufacturing," in *Proc. 12th Conf. Learn. Factories (CLF)*, Singapore (2022), doi: 10.2139/ssrn.4073942.

[9] Fraunhofer IPA, "Dependable AI – Using AI in safety-critical industrial applications," *Fraunhofer Inst. Manuf. Eng. Autom. (IPA)*, Stuttgart, Germany, White Paper (2021).

[10] S. Tripathi, D. Muhr, M. Brunner, H. Jodlbauer, M. Dehmer, and F. Emmert-Streib, "Ensuring the robustness and reliability of data-driven knowledge discovery models in production and manufacturing," *Front. Artif. Intell.*, vol. 4, Art. no. 576892 (2021).

[11] I. Žliobaitė, M. Pechenizkiy, and J. Gama, "An overview of concept drift applications," in *Big Data Analysis: New Algorithms for a New Society,* Studies in Big Data, vol. 16. Cham, Switzerland: Springer, pp. 91–114 (2016).

[12] N. Jourdan, S. Sen, E. J. Husom, E. Garcia-Ceja, T. Biegel, and J. Metternich, "On the reliability of machine learning applications in manufacturing environments," in *Proc. NeurIPS Workshop Distribution Shifts: Connecting Methods and Applications*, arXiv:2112.06986 (2021).

[13] S. Ackerman, E. Farchi, O. Raz, M. Zalmanovici, and P. Dube, "Detection of data drift and outliers affecting machine learning model performance over time," arXiv:2012.09258 (2020).

[14] E. S. Page, "Continuous inspection schemes," *Biometrika,* vol. 41, no. 1/2, pp. 100–115 (1954).

[15] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Proc. Brazilian Symp. Artif. Intell.*, Springer, pp. 286–295 (2004).

[16] I. Žliobaitė, "Learning under concept drift: An overview," arXiv:1010.4784 (2010).

[17] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 12, pp. 2346–2363 (2019), doi: 10.1109/TKDE.2018.2876857.

[18] S. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. Chan, "Cost-based modeling for fraud and intrusion detection: Results from the JAM project," in *Proc. DARPA Inf. Survivability Conf.* (2000).

[19] M. Baena-García, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavaldà, and R. Morales-Bueno, "Early drift detection method," in *Proc. 4th Int. Workshop Knowl. Discovery Data Streams*, vol. 6, pp. 77–86 (2006).

[20] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," in *Proc. SIAM Int. Conf. Data Mining*, pp. 443–448 (2007).

[21] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *Int. J. Math. Models Methods Appl. Sci.*, vol. 1, no. 4, pp. 300–307 (2007).

[22] S. Studer, T. B. Bui, C. Drescher, A. Hanuschkin, L. Winkler, S. Peters, and K.-R. Müller, "Towards CRISP-ML(Q): A machine learning process model with quality assurance methodology," *Mach. Learn. Knowl. Extr.*, vol. 3, no. 2, pp. 392–413 (2021).

[23] R. Wirth and J. Hipp, "CRISP-DM: Towards a standard process model for data mining," in *Proc. 4th Int. Conf. Practical Appl. Knowl. Discovery Data Mining,* Manchester, U.K., vol. 1, pp. 29–39 (2000).

[24] M. Chui, B. Hall, A. Singla, and A. Sukharevsky, "The state of AI in 2021," McKinsey & Company, Tech. Rep. (2021).

[25] M. Chui, B. Hall, H. Mayhew, A. Singla, and A. Sukharevsky, "The state of AI in 2022—and a half decade in review," *QuantumBlack by McKinsey, Tech. Rep.* (2022).

[26] M. A. Tnani, M. Feil, and K. Diepold, "Smart data collection system for brownfield CNC milling machines: A new benchmark dataset for data-driven machine monitoring," *Procedia CIRP,* vol. 107, pp. 131–136 (2022), doi: 10.1016/j.procir.2022.04.022.

[27] S. Fahle, C. Prinz, and B. Kuhlenkötter, "Systematic review on machine learning (ML) methods for manufacturing processes—Identifying artificial intelligence (AI) methods for field application," *Procedia CIRP,* vol. 93, pp. 413–418 (2020).

[28] C.-C. Lin, D.-J. Deng, C.-H. Kuo, and L. Chen, "Concept drift detection and adaption in big imbalance industrial IoT data using an ensemble learning method of offline classifiers," *IEEE Access,* vol. 7, pp. 56198–56207 (2019).

[29] L. Baier, T. Schlör, J. Schöffer, and N. Kühl, "Detecting concept drift with neural network model uncertainty,"arXiv:2107.01873 (2021).

[30] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn., PMLR,* pp. 1050–1059 (2016).

[31] B. X. Yong, Y. Fathy, and A. Brintrup, "Bayesian autoencoders for drift detection in industrial environments," in *Proc. IEEE Int. Workshop Metrol. Ind. 4.0 IoT (MetroInd4.0&IoT)*, pp. 627–631 (2020), doi: 10.1109/MetroInd4.0IoT48571.2020.9138306.

[32] F. Bachinger and G. Kronberger, "Concept for a technical infrastructure for management of predictive models in industrial applications," in *Computer Aided Systems Theory – EUROCAST 2019, Lecture Notes in Computer Science*, vol. 12013. Cham, Switzerland: Springer, pp. 263–270 (2020), doi: 10.1007/978-3-030-45093-9_32.

[33] F. Bachinger, G. Kronberger, and M. Affenzeller, "Continuous improvement and adaptation of predictive models in smart manufacturing and model management," *IET Collab. Intell. Manuf.,* vol. 3, no. 1, pp. 48–63 (2021), doi: 10.1049/cim2.12009.

[34] J. Zenisek, F. Holzinger, and M. Affenzeller, "Machine learning based concept drift detection for predictive maintenance," *Comput. Ind. Eng.,* vol. 137, Art. no. 106031 (2019).

[35] S. Gregor and A. R. Hevner, "Positioning and presenting design science research for maximum impact," *MIS Q.,* vol. 37, no. 2, pp. 337–355 (2013).

[36] B. Sahiner, W. Chen, R. K. Samala, and N. Petrick, "Data drift in medical machine learning: Implications and potential remedies," *Br. J. Radiol.,* vol. 96, no. 1150, Art. no. 20220878 (2023), doi: 10.1259/bjr.20220878.

[37] J. De Fauw, J. R. Ledsam, B. Romera-Paredes, S. Nikolov, N. Tomasev, S. Blackwell, H. Askham, X. Glorot, B. O'Donoghue, D. Visentin, G. van den Driessche, B. Lakshminarayanan, C. Meyer, F. Mackinder, S. Bouton, K. Ayoub, R. Chopra, D. King, A. Karthikesalingam, C. O. Hughes, R. Raine, J. Hughes, D. A. Sim, C. Egan, A. Tufail, H. Montgomery, D. Hassabis, G. Rees, T. Back, P. T. Khaw, M. Suleyman, J. Cornebise, P. A. Keane, and O. Ronneberger, "Clinically applicable deep learning for diagnosis and referral in retinal disease," *Nat. Med.,* vol. 24, no. 9, pp. 1342–1350 (2018).

[38] J. R. Zech, M. A. Badgeley, M. Liu, A. B. Costa, J. J. Titano, and E. K. Oermann, "Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study," *PLOS Med.,* vol. 15, no. 11, Art. no. e1002683 (2018).

[39] G. Mårtensson, D. Ferreira, T. Granberg, L. Cavallin, K. Oppedal, A. Padovani, I. Rektorova, L. Bonanni, M. Pardini, M. G. Kramberger, J. P. Taylor, J. Hort, J. Snædal, J. Kulisevsky, F. Blanc, A. Antonini, P. Mecocci, B. Vellas, M. Tsolaki, I. Kłoszewska, H. Soininen, S. Lovestone, A. Simmons, D. Aarsland, and E. Westman, "The reliability of a deep learning model in clinical out-of-distribution MRI data: A multicohort study," *Med. Image Anal.*, vol. 66, Art. no. 101714 (2020).

[40] B. Nestor, M. B. A. McDermott, W. Boag, G. Berner, T. Naumann, M. C. Hughes, A. Goldenberg, and M. Ghassemi, "Feature robustness in non-stationary health records: Caveats to deployable model performance in common clinical machine learning tasks," in *Proc. Mach. Learn. Healthcare Conf. (MLHC),* PMLR, vol. 106 (2019).

[41] A. J. Larrazabal, N. Nieto, V. Peterson, D. H. Milone, and E. Ferrante, "Gender imbalance in medical imaging datasets produces biased

classifiers for computer-aided diagnosis," *Proc. Natl. Acad. Sci.* U.S.A., vol. 117, no. 23, pp. 12592–12594 (2020).

[42] C. Duckworth, F. P. Chmiel, D. K. Burns, Z. D. Zlatev, N. M. White, T. W. V. Daniels, M. Kiuber, and M. J. Boniface, "Using explainable machine learning to characterise data drift and detect emergent health risks for emergency department admissions during COVID-19," *Sci. Rep.,* vol. 11, no. 1, Art. no. 23017 (2021).

[43] S. M. Faizanut Tauhid, S. Tanweer, M. T. Nafis, M. A. Ahad, and S. M. F. Malik, "Machine learning-driven data analytics for improved diagnostic accuracy, treatment efficacy, and real-time monitoring in smart health care," *J. Inf. Optim. Sci.,* vol. 46, no. 7, pp. 2291–2317 (2025), doi: 10.47974/JIOS-2031.

[44] G. S. Shanmugam, A. Kunwar, M. Anbarasi, O. Nahta, V. Kumar, and D. Jain, "Detailed predictive modeling of hospital readmission risk using a class-balanced machine learning approach," *J. Inf. Optim. Sci.,* vol. 47, no. 9-B, pp. 3509–3519 (2026), doi: 10.47974/JIOS-2448.

Received July, 2025