Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

StrokeGuard: Cloud-Deployed Calibrated-MLP with K-Means Boundary Undersampling for Scalable Stroke Risk Prediction

Mohamed Shaban
Egypt-Japan University of Science and Technology
Shahd Ibrahim
Egypt-Japan University of Science and Technology
Manal M. Ghaith
Egypt-Japan University of Science and Technology
Ahmed M. Anter
Ahmed_Anter@fcis.bsu.edu.eg

Egypt-Japan University of Science and Technology

Research Article

Keywords:

Posted Date: November 2nd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11286884/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

StrokeGuard: Cloud-Deployed Calibrated-MLP with K-Means Boundary Undersampling for Scalable Stroke Risk Prediction

Mohamed Shaban*, Shahd Ibrahim*, Manal M. Ghaith*, and Ahmed M. Anter*†
*Egypt-Japan University of Science and Technology (E-JUST), Alexandria, 21934, Egypt
‡Faculty of Computers and Artificial Intelligence, Beni-Suef University, Benisuef, 62511, Egypt
Corresponding Author: Ahmed M. Anter, Email: ahmed_anter@fcis.bsu.edu.eg

***Abstract*—Stroke accounts for roughly one in nine deaths worldwide and ranks among the most disabling non-communicable diseases globally. Timely identification of at-risk individuals is clinically actionable, yet two persistent obstacles hamper the deployment of automated screening tools: (1) the pronounced scarcity of stroke-positive records in routine clinical databases, where positive cases often constitute less than 5% of the cohort and yield imbalance ratios that can exceed 19:1; and (2) the near-universal absence of validated, production-grade inference infrastructure in published stroke-ML research. We present *StrokeGuard*, an end-to-end system that couples a novel K-Means cluster-based undersampling strategy with a threshold-calibrated multi-layer perceptron (MLP) and delivers the trained model to a multi-region cloud platform with fully benchmarked latency, throughput, and cost characteristics. The undersampling algorithm divides the PCA-transformed feature space into four clusters, then draws majority-class records whose proximity to the local minority centroid is smallest, thereby concentrating training signal on the class boundary while reducing the imbalance ratio from 19.6:1 to 5:1 without generating synthetic data. The MLP is optimized with class-weighted binary cross-entropy, and its decision boundary is shifted to $\tau = 0.3$ to maximize G-Mean. Preprocessing comprises Random Forest regression for missing BMI imputation ($R^2 = 0.707$), z-score normalization using training-set statistics, and Principal Component Analysis retaining 89% of total variance. Evaluated on 45,108 clinical records (2,631 held-out test samples), the system achieves 92.51% accuracy, 72.60% recall, 80.51% precision, 96.49% specificity, 83.70% G-Mean, 76.35% F1 score, and AUC-ROC = 0.8706, outperforming seven baseline classifiers. Five-fold cross-validation yields $92.36\% \pm 0.31\%$, and the train-to-test accuracy gap is 1.61%, confirming generalization. Cloud deployment across US East, EU West, and Asia Pacific produces mean inference latency of 18.72 ms (P95: 24.34 ms; P99: 34.24 ms) and throughput up to 54.7 req/s, with auto-scaling from 1 to 8 instances at linear cost ($0.115 to $0.920/hr). All three regions sustain sub-40 ms P99 latency, satisfying real-time clinical screening requirements.**

***Index Terms*—Auto-scaling, class imbalance, cloud computing, K-Means undersampling, real-time inference, stroke prediction.**

I. Introduction

STROKE is a sudden disruption of cerebral blood flow that deprives neural tissue of oxygen, triggering rapid cell death. The World Health Organization classifies it as the second leading cause of death globally [1]. Epidemiological analyses estimate that tens of millions of incident stroke events occur each year worldwide; approximately one-third prove fatal and an additional third result in chronic functional impairment, imposing a dual burden of premature mortality and long-term disability [2]. In the United States alone, direct and indirect economic losses attributable to stroke exceed $50 billion annually [3].

When vascular risk is identified early, preventive action—lifestyle counseling, pharmacological management, and heightened clinical surveillance—can meaningfully curtail both incidence and case fatality [4]. Machine learning (ML) models trained on electronic health records have demonstrated capacity to integrate heterogeneous demographic, physiological, and behavioral signals in ways that conventional logistic risk scores may not fully exploit [5].

Two structural problems nonetheless impede the clinical translation of these models.

Severe Class Imbalance. In community-drawn and hospital stroke registries, stroke-positive entries typically account for fewer than 5% of all records, yielding majority-to-minority ratios that approach or exceed 19:1 in large cohorts [6]. Classifiers optimized on such skewed distributions tend to favor the dominant class, reaching high headline accuracy while producing unacceptably low sensitivity for the rare positive class. Conventional remedies each carry trade-offs. Synthetic oversampling via SMOTE [25] creates new minority-class points by linear interpolation within the minority neighborhood; in medical data this can yield feature combinations—such as merging the clinical profile of a young normotensive patient with that of an elderly diabetic—that have no counterpart in observed physiology, thereby introducing medically implausible noise into the training distribution. Random undersampling, by contrast, discards majority records without any spatial criterion, so examples that lie in contested regions of the feature space—precisely those that carry the most information about the class boundary—are lost at the same rate as obviously non-stroke cases far from that boundary [7].

Deployment Infrastructure Gap. The overwhelming majority of published stroke-ML studies terminate at offline evaluation. Production-grade clinical use additionally demands sub-100 ms inference for interactive tools, geographic distribution with local data-residency compliance, high availability, and elastic capacity to absorb variable patient volumes [8], [9]. To our knowledge, few if any prior studies have jointly reported percentile latency distributions (P95, P99), throughput

profiles under variable concurrent load, multi-region inference times, and cost trajectories across auto-scaling thresholds for a stroke-specific ML pipeline.

This paper addresses both problems through four principal contributions:

- **K-Means cluster-based undersampling.** The PCA-transformed feature space is partitioned into $k = 4$ clusters; within each cluster, majority-class records are ranked by Euclidean distance to the local minority centroid, and the $k_s \cdot |C_j^{\min}|$ nearest are retained. This concentrates retained majority-class diversity along the class boundary while eliminating redundant, easily classified records from cluster interiors, reducing the imbalance ratio from 19.6:1 to 5:1 without synthetic data generation.
- **Integrated preprocessing and classification pipeline.** Random Forest regression for BMI imputation ($R^2 = 0.707$, outperforming mean, median, and KNN baselines); PCA-based dimensionality reduction at 89% variance retention; MLP training with class-weighted loss; and threshold optimization at $\tau = 0.3$. The pipeline yields 92.51% accuracy, 72.60% recall, and 83.70% G-Mean on the held-out test set, exceeding all seven compared classifiers on balanced metrics.
- **Multi-region cloud deployment with full benchmarking.** A model registry, RESTful inference API, monitoring stack, and auto-scaling policy are deployed across three geographic zones. Benchmarks cover P95/P99 latency, throughput under 10–1,000 concurrent requests, cross-region comparison, and cost scaling.
- **Systematic feature-reduction comparison.** Five methods (PCA, SVD, RFE, LASSO, ElasticNet) are evaluated on an identical train-test split, establishing that PCA at 89% variance retention matches the no-reduction ceiling and dominates all wrapper and embedded alternatives.

Section II reviews related work in stroke prediction and healthcare ML deployment. Section III details each algorithmic component. Section IV describes the dataset and pipeline. Section V reports experimental results. Section VI concludes and outlines future directions.

II. Related Work

Stroke-prediction research can be organized around three methodological axes: the strategy used to correct class imbalance, the classifier architecture, and the extent to which the resulting model is operationally validated. We survey each axis and identify unaddressed combinations that motivate the present work.

A. Correcting Class Imbalance

Stroke-positive cases constitute roughly 4–6% of most clinical cohorts, creating imbalance ratios near 19–20:1. Two correction paradigms dominate the literature.

Dev and Wang [4] applied SMOTE to generate synthetic stroke samples and trained a stacked ensemble (logistic regression, SVM, random forest), reporting 95% accuracy without disclosing recall or G-Mean, so minority-class sensitivity cannot be verified. Tazin et al. [7] likewise coupled SMOTE with a random forest, attaining 96% accuracy. Interpolation-based oversampling operates within the manifold of existing stroke cases, but in tabular clinical data the interpolated region does not always correspond to physiologically coherent patient profiles, introducing distribution mismatch between synthetic and real cases.

Random undersampling eliminates majority records until a target ratio is reached. Sailasya and Kumari [8] used random forest on an unbalanced set (82% accuracy), substantially below SMOTE-corrected models. A key limitation of random undersampling is that it assigns equal removal probability to all majority records irrespective of their location in feature space, offering no mechanism to preserve the records most relevant to boundary learning.

Cluster-based undersampling partitions the feature space and draws majority records near minority cluster centers. Lin et al. [26] applied this to UCI benchmark datasets with SVM classifiers, demonstrating improved F1 scores over random undersampling. Liu et al. [27] extended the approach with adaptive cluster counts. Neither study evaluated on medical data or combined the strategy with neural networks. The CDF2S system [31] applied cluster undersampling to stroke prediction with a deep-forest classifier (AUC 0.94) but did not explore MLP architectures, threshold calibration, or deployment.

B. Classifier Architectures

Tree ensembles remain predominant. Dritsas and Trigka [5] compared seven supervised learners and found cross-validated stacking reached AUC 0.93. Liu et al. [6] used Bayesian hyperparameter search with XGBoost on hospital records ($n = 3{,}847$), obtaining 96.5% accuracy. Hung et al. [10] combined gradient boosting with a neural network (AUC 0.94) at the cost of 18 GB RAM and 4 hours of training time, constraining replication in resource-limited settings.

Deep learning approaches show variable returns. Cheon et al. [9] applied LSTM to time-series EHR data (83.5% recall) but required at least five sequential clinic visits, excluding 40% of the cohort. Sharma and Chaurasia [12] trained a class-weighted deep neural network (85% recall, 91% accuracy) without combining weighting with structured undersampling or evaluating competing feature representations.

Extreme Learning Machines (ELMs) have been proposed for imbalanced medical classification. Zong et al. [23] introduced weighted ELM with inverse class-frequency weights; Zhang et al. [24] developed an adaptive variant whose hidden-layer size adjusts to per-class sample counts. These methods attain 75–80% recall on standard imbalanced benchmarks but have not been systematically compared against cluster-undersampled MLPs on stroke data, a gap this paper addresses.

C. Feature Selection and Reduction

Most stroke studies use all available features or apply a single reduction method without cross-method comparison. Alqahtani et al. [15] used SHAP with random forest and XGBoost for post-hoc explanation (94.7% accuracy), but SHAP was not used as a prospective filter during training.

Wrapper and embedded methods (RFE, LASSO) have been benchmarked on other medical classification tasks [29], [30], yet no stroke study to our knowledge has compared PCA, SVD, RFE, LASSO, Ridge, and ElasticNet on a shared train-test partition.

D. Deployment in Clinical Systems

Operational deployment remains the least explored dimension. Govindarajan et al. [13] demonstrated REST-API delivery of a logistic regression stroke tool on AWS (88% accuracy) but provided no latency, throughput, or scaling measurements. Islam et al. [14] proposed an IoT edge architecture for ECG-based detection (93% accuracy); edge deployments inherently restrict geographical reach, serving only patients physically connected to each edge device.

To our knowledge, no prior study has simultaneously reported P95/P99 latency distributions, concurrent-load throughput curves, cross-region inference comparisons, and auto-scaling cost trajectories for a stroke-prediction ML system.

E. Research Gaps

Table I maps methodological coverage across representative works. Key unaddressed combinations are: (i) cluster-based undersampling paired with threshold-calibrated MLPs, (ii) systematic cross-method feature-reduction comparison on stroke data, and (iii) full deployment benchmarking with latency, throughput, and cost metrics. StrokeGuard addresses all three.

III. Methodology

This section describes the algorithmic components of the proposed pipeline. Each subsection presents the technique, its mathematical formulation, and the rationale for its role in stroke risk prediction.

A. Missing Value Imputation via Random Forest Regression

The dataset contains missing BMI values in 3.9% of records. Global substitution strategies such as mean imputation ($\hat{x}_j = \bar{x}_j$) or median imputation ($\hat{x}_j = \text{median}(x_j)$) treat BMI as independent of all other covariates, discarding the correlation structure between BMI and features such as age, glucose level, and hypertension status. We instead model each missing BMI value as a function of the patient's observed covariates via Random Forest regression [16].

For each record i whose BMI is unobserved, the imputed value is

$$\hat{x}_j^{(i)} = \frac{1}{T}\sum_{t=1}^{T} f_t\left(x_{-j}^{(i)}\right), \tag{1}$$

where f_t is the t-th regression tree, $T = 50$ is the ensemble size, and $x_{-j}^{(i)}$ collects all features except BMI for record i. Each tree is grown to maximum depth 8 on a bootstrap draw of records with observed BMI. Bootstrap aggregation smooths individual-tree variance; depth 8 permits nonlinear interactions (e.g., the BMI–glucose relationship differs substantially between younger and older patients) without overfitting individual training records.

We fixed $T = 50$ and depth 8 after observing that larger ensembles ($T > 100$) improved R^2 by fewer than 0.01 while increasing imputation time proportionally, and that shallower trees failed to capture nonlinear interactions critical for older, multi-morbid subgroups.

B. Feature Normalization by Z-Score Standardization

Gradient-based optimization and distance metrics are sensitive to the scale of input features. In the present dataset, average glucose spans 55–271 mg/dL while binary risk-factor indicators occupy $\{0, 1\}$; without normalization, high-magnitude features dominate both Euclidean distances (used in K-Means clustering and KNN imputation) and weight-gradient magnitudes during backpropagation.

We apply z-score standardization, transforming each feature x_j as

$$z_j^{(i)} = \frac{x_j^{(i)} - \mu_j}{\sigma_j}, \tag{2}$$

where μ_j and σ_j are the mean and standard deviation of feature j estimated exclusively from the training partition. The identical training-set parameters are subsequently applied to the test partition, ensuring that no information from the test distribution influences the normalization step—a design choice that prevents a specific form of data leakage that can inflate reported performance [34]. After transformation, every feature contributes on a comparable scale to distance calculations and gradient updates, improving convergence stability and reducing sensitivity to random weight initialization.

C. Dimensionality Reduction via Principal Component Analysis

PCA [17] maps a p-dimensional standardized feature vector $\mathbf{x}$ onto the $d < p$ orthogonal directions that capture the greatest share of total variance. Let $\mathbf{C} \in \mathbb{R}^{p\times p}$ be the sample covariance matrix of the standardized training features. Its spectral decomposition is $\mathbf{C} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$, where columns of $\mathbf{V}$ are eigenvectors (principal directions) and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_p)$ with $\lambda_1 \geq \cdots \geq \lambda_p \geq 0$.

Projecting onto the leading d eigenvectors gives $\mathbf{z} = \mathbf{V}_d^T\mathbf{x}$, and the retained variance fraction is

$$\rho(d) = \frac{\sum_{i=1}^{d}\lambda_i}{\sum_{i=1}^{p}\lambda_i}. \tag{3}$$

We choose d so that $\rho(d) \geq 0.89$. A higher threshold of 0.95 retained all ten components (no reduction); a lower threshold of 0.85 discarded components with genuine discriminative content, degrading test accuracy by 3.5 percentage points. At $\rho = 0.89$, nine components are retained; the discarded component captures primarily noise rather than stroke-relevant variance, yielding a mild regularization effect alongside the dimensionality reduction.

TABLE I: Methodological Gaps in Stroke-Prediction Literature.

Ref	Authors	Year	Imbalance Method	Best Metric	Missing Components
[4]	Dev & Wang	2022	SMOTE	Acc: 95.0%	Recall/G-Mean unreported; no deployment
[5]	Dritsas & Trigka	2022	None	AUC: 0.93	No imbalance correction; offline only
[6]	Liu et al.	2023	Random oversample	Acc: 96.5%	Small n; batch only; no real-time test
[7]	Tazin et al.	2021	SMOTE	Acc: 96.0%	Synthetic-sample risk; no boundary focus
[8]	Sailasya & Kumari	2021	None	Acc: 82.0%	No imbalance correction
[9]	Cheon et al.	2019	None	Recall: 83.5%	Excludes 40% patients; no scalability
[31]	CDF2S	2024	Cluster undersample	AUC: 0.94	No MLP; no threshold calibration; no deployment
[13]	Govindarajan et al.	2020	None	Acc: 88.0%	Simple LR; no latency/cost data
	This study	**2026**	**K-Means + MLP + threshold**	**Acc: 92.51%, G-M: 83.70%**	**Addresses all gaps**

D. K-Means Cluster-Based Undersampling

Uniform random undersampling selects n_{target} majority records at random from the n_{maj} available, treating every majority record as equally expendable. Records situated close to the minority class in feature space—those that define the region where the classifier must be most discriminating—are removed at the same rate as records far from any stroke case. This wastes the most informative segment of the majority class.

Our algorithm partitions the PCA-transformed space into k clusters via K-Means [18] and then draws majority records from each cluster according to their Euclidean proximity to the local minority centroid.

Let $C_1, \ldots, C_k$ denote the cluster assignments. For cluster C_j, define

$$C_j^{\min} = \{i \in C_j : y_i = 1\}, \quad C_j^{\text{maj}} = \{i \in C_j : y_i = 0\}. \quad (4)$$

The minority centroid within cluster j is

$$\boldsymbol{\mu}_j = \frac{1}{|C_j^{\min}|} \sum_{i \in C_j^{\min}} \mathbf{x}_i. \quad (5)$$

Each majority record $i \in C_j^{\text{maj}}$ receives a proximity score

$$d_i = \|\mathbf{x}_i - \boldsymbol{\mu}_j\|_2. \quad (6)$$

We retain the $n_j = k_s \cdot |C_j^{\min}|$ majority records with the smallest d_i values. The final undersampled majority class is the union of all $I_{\text{sel}} = \bigcup_j \{n_j \text{ nearest in } C_j^{\text{maj}}\}$.

Setting $k = 4$ provides enough spatial granularity to identify distinct boundary regions without over-fragmenting the minority class. The sampling ratio $k_s = 5$ produces a 5:1 training imbalance (10,960 majority, 2,192 minority), sufficient for the classifier to learn the minority manifold while retaining boundary-relevant diversity in the majority class. Critically, no synthetic data are created; every retained record corresponds to an actual clinical observation.

E. MLP Architecture and Training Objective

The classifier is a feedforward network with two hidden layers of widths 64 and 32. Denoting the input (9 PCA components) as $h_0 = \mathbf{x}$, the first hidden layer computes

$$h_1 = \text{ReLU}(W_1 h_0 + b_1), \quad W_1 \in \mathbb{R}^{64\times 9},\ b_1 \in \mathbb{R}^{64}. \quad (7)$$

ReLU ($\text{ReLU}(z) = \max(0, z)$) introduces piecewise-linear nonlinearity while avoiding the gradient saturation that plagues sigmoid and tanh activations in intermediate layers. The second hidden layer is

$$h_2 = \text{ReLU}(W_2 h_1 + b_2), \quad W_2 \in \mathbb{R}^{32\times 64},\ b_2 \in \mathbb{R}^{32}. \quad (8)$$

The output layer maps to a stroke probability via

$$\hat{y} = \sigma(W_3 h_2 + b_3) = \frac{1}{1 + \exp(-(W_3 h_2 + b_3))}, \quad W_3 \in \mathbb{R}^{1\times 32}. \quad (9)$$

Training minimizes a class-weighted binary cross-entropy objective:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} w_{y_i} \left[y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \right], \quad (10)$$

where w_c is set inversely proportional to class c's frequency in the training partition. For the 5:1 balanced set, $w_0 = 1.0$ and $w_1 = 5.0$, imposing a five-fold higher gradient penalty for missed stroke cases than for false alarms. This provides a second layer of imbalance correction supplementary to the undersampling step.

Parameters are updated with the Adam optimizer [20] ($\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$), whose per-parameter adaptive learning rates accelerate convergence relative to vanilla SGD. A 15% within-training validation split monitors loss after each epoch; training halts if validation loss fails to improve for 10 consecutive epochs (patience), with an absolute ceiling of 200 epochs. In practice, convergence occurs between epochs 80 and 120.

F. Classification Threshold Calibration

The trained MLP outputs a calibrated probability $\hat{y} \in [0, 1]$. Under the default decision boundary $\hat{c} = \mathbf{1}[\hat{y} > 0.5]$, a false negative (an undetected stroke) and a false positive (unnecessary follow-up for a healthy patient) are penalized symmetrically—an assumption that is clinically inappropriate when the cost of missing a stroke event substantially exceeds the cost of a precautionary referral.

We shift the boundary to $\tau = 0.3$:

$$\hat{c} = \begin{cases} 1 & \text{if } \hat{y} > 0.3, \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

The threshold was selected by sweeping $\tau \in [0.1, 0.9]$ on a held-out validation fold and choosing the value that maximized the G-Mean, which balances sensitivity across both class directions:

$$\text{G-Mean} = \sqrt{\text{Recall} \times \text{Specificity}}. \tag{12}$$

At $\tau = 0.3$, the model simultaneously achieves 72.60% recall and 96.49% specificity, giving G-Mean = 83.70%.

G. Evaluation Metrics

We report six complementary metrics derived from the confusion matrix elements (TP, TN, FP, FN):

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}, \tag{13}$$

$$\text{Recall} = \frac{TP}{TP+FN}, \tag{14}$$

$$\text{Precision} = \frac{TP}{TP+FP}, \tag{15}$$

$$\text{Specificity} = \frac{TN}{TN+FP}, \tag{16}$$

$$\text{G-Mean} = \sqrt{\text{Recall} \times \text{Specificity}}, \tag{17}$$

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{18}$$

Accuracy alone is an unreliable indicator for skewed test sets: a degenerate classifier that labels every record as non-stroke achieves over 95% accuracy on this dataset while attaining zero recall. G-Mean is particularly suited to imbalanced problems because it rewards high performance on both the minority and majority class simultaneously; a model with high overall accuracy but near-zero sensitivity will be penalized with a correspondingly low G-Mean [33]. F1 is the harmonic mean of precision and recall, penalizing classifiers that sacrifice either dimension.

We additionally report the Area Under the Receiver Operating Characteristic Curve (AUC-ROC), which aggregates discriminative ability across all possible decision thresholds and is therefore independent of the chosen operating point τ [32].

IV. Dataset and Pipeline

The stroke prediction system integrates preprocessing, undersampling, classification, and cloud deployment into a single end-to-end workflow. Figure 1 illustrates the complete architecture from raw input to multi-region inference.

A. Dataset Characteristics

We use the Synthetic Stroke Prediction Dataset [21], which contains 50,000 records and 12 fields. After removing the identifier column, one-hot encoding categorical variables (gender, marital status, work type, residence type, smoking status), and filtering physiologically implausible records (age < 25; BMI > 60), 45,108 samples and 10 predictive features remain. Table II summarizes sample counts at each processing stage.

TABLE II: Dataset Summary at Each Processing Stage.

Property	Value
Original samples	50,000
Original fields (incl. id, target)	12
Samples after outlier filtering	45,108
Predictive features (post-encoding)	10
PCA components retained	9
Class 0 (No Stroke, original)	42,916
Class 1 (Stroke, original)	2,192
Original imbalance ratio	19.6:1
Class 0 (post K-Means undersampling)	10,960
Class 1 (all preserved)	2,192
Total balanced samples	13,152
Balanced imbalance ratio	5.0:1
Training samples (80%)	10,521
Test samples (20%)	2,631

The original dataset contains 42,916 non-stroke records (95.14%) and 2,192 stroke records (4.86%), yielding an imbalance ratio of 19.6:1.

1) Descriptive Statistics: Table III reports descriptive statistics for continuous features after outlier filtering. The pronounced scale differences across features (age variance: 217.86; glucose variance: 2082.10; BMI variance: 61.15) confirm the necessity of z-score standardization before distance- and gradient-based operations.

TABLE III: Descriptive Statistics of Continuous Features.

Statistics	Age	Glucose	BMI
Min	25.00	55.12	10.30
Max	82.00	271.74	60.00
Mean	51.23	109.87	29.54
Std Dev	14.76	45.63	7.820
Variance	217.86	2082.1	61.15
Skewness	0.120	1.240	0.450
Kurtosis	−0.870	0.580	0.210

The feature set spans demographic variables (age, gender), clinical history (hypertension, heart disease), lifestyle and occupational characteristics (smoking status, work type, residence type), and physiological measurements (BMI, average glucose level).

B. Preprocessing Steps

Three operations are applied sequentially before any model training.

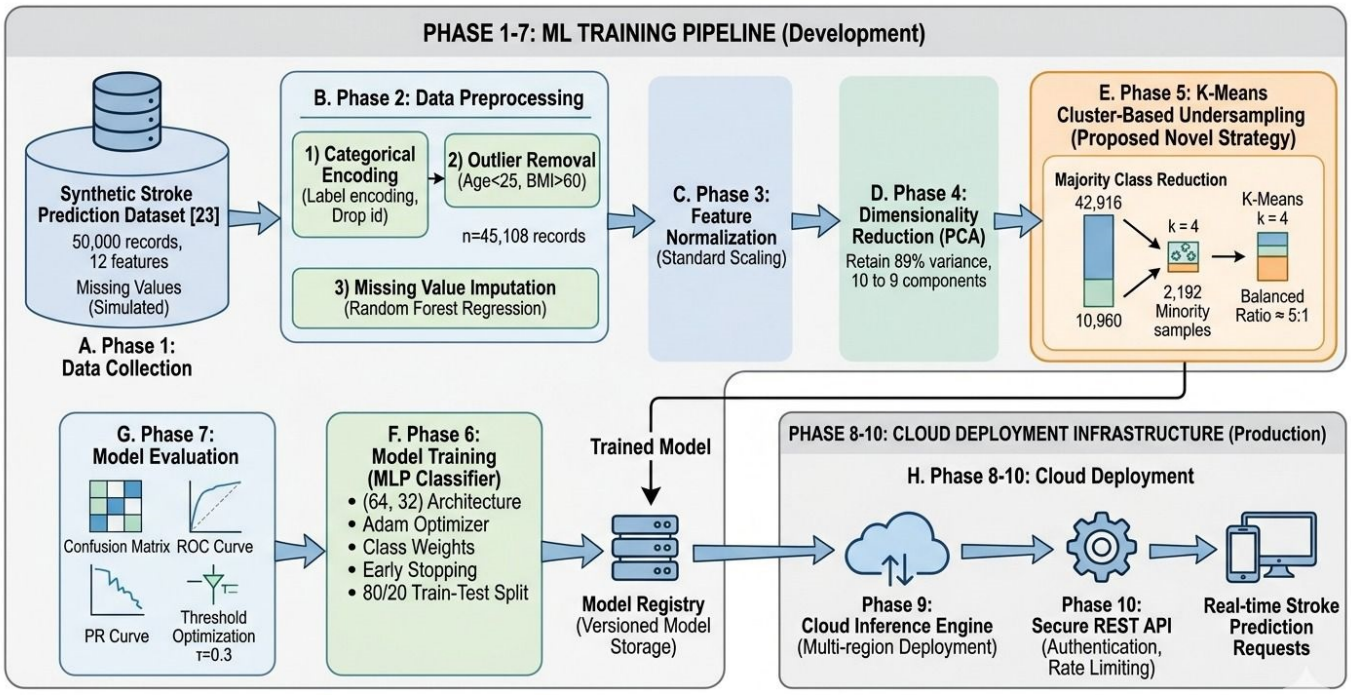


Fig. 1: End-to-end StrokeGuard pipeline, from raw clinical data to multi-region cloud inference. Preprocessing, undersampling, model training, and deployment are integrated into a single workflow.

Categorical Encoding. Categorical variables—gender, marital status, work type, residence type, and smoking status—are mapped to integer codes via label encoding. The identifier column is excluded because it carries no clinical information.

Outlier Filtering. Analysis is restricted to adult patients (age ≥ 25) with plausible BMI values (BMI ≤ 60), reducing the cohort from 50,000 to 45,108 records.

Missing Value Imputation. BMI is absent for 3.9% of records. Imputed values are produced by the Random Forest regressor described in Section III-A. Table IV compares four methods on a validation subset with artificially introduced missingness. Random Forest achieves the highest R^2 (0.707) and lowest RMSE (4.23).

TABLE IV: BMI Imputation Method Comparison.

Method	RMSE	R^2
Mean Imputation	7.82	0.000
Median Imputation	7.79	0.005
KNN Imputation ($k = 5$)	5.41	0.521
RF Imputation	**4.23**	**0.707**

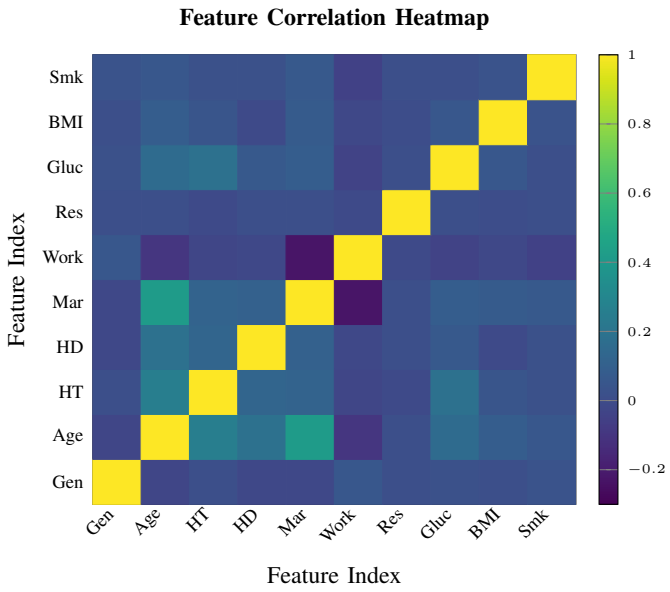


Fig. 2: Pearson correlation matrix. Age, hypertension, and average glucose level exhibit the strongest associations with stroke outcome.

C. Feature Correlation Analysis

Pearson correlation analysis (Fig. 2) reveals that age exhibits the strongest positive association with stroke occurrence ($r = 0.25$), followed by average glucose level ($r = 0.15$) and hypertension ($r = 0.13$). These relationships are consistent with established clinical risk-factor hierarchies [2].

D. Normalization and Dimensionality Reduction

Z-score standardization (Section III-B) is fitted on the training partition and applied to both partitions. PCA (Section III-C) then maps the 10-dimensional normalized space onto 9 principal components representing 89% of cumulative variance. The 89% threshold was determined empirically: the 95% setting preserved all ten features, while the 85%

setting discarded features with discriminative content, reducing accuracy by 3.5 percentage points.

E. K-Means Cluster-Based Undersampling

Algorithm 1 formalizes the undersampling procedure. With $k = 4$ and $k_s = 5$, the algorithm retains all 2,192 minority records and reduces the majority class from 42,916 to 10,960 records, yielding 13,152 balanced training samples at a 5:1 ratio.

Algorithm 1 K-Means Cluster-Based Undersampling.

Require: PCA features X, labels y, clusters $k = 4$, ratio $k_s = 5$
Ensure: Balanced dataset indices I_{bal}
Apply K-Means with k clusters to X
$I_{\min} \leftarrow \{i : y_i = 1\}$ {Stroke records}
$I_{\text{maj}} \leftarrow \{i : y_i = 0\}$ {Non-stroke records}
$I_{\text{sel}} \leftarrow \emptyset$
for each cluster C_j, $j = 1, \ldots, k$ **do**
  $C_j^{\min} \leftarrow I_{\min} \cap C_j$
  $C_j^{\text{maj}} \leftarrow I_{\text{maj}} \cap C_j$
  if $C_j^{\min} \neq \emptyset$ **and** $C_j^{\text{maj}} \neq \emptyset$ **then**
    $\boldsymbol{\mu}_j \leftarrow \text{mean}(X[C_j^{\min}])$
    Compute $d_i = \|X_i - \boldsymbol{\mu}_j\|_2$ for all $i \in C_j^{\text{maj}}$
    $n \leftarrow \min(k_s \cdot |C_j^{\min}|, |C_j^{\text{maj}}|)$
    Select n majority records with smallest d_i
    $I_{\text{sel}} \leftarrow I_{\text{sel}} \cup \{\text{selected indices}\}$
  end if
end for
$I_{\text{bal}} \leftarrow I_{\min} \cup I_{\text{sel}}$
return I_{bal}

F. MLP Training Configuration

The MLP (64, 32 hidden units; ReLU activation; sigmoid output) is trained on the 13,152-sample balanced dataset with the following settings:

- Class weights: $w_0 = 1.0$ (majority), $w_1 = 5.0$ (minority), derived from the inverse class-frequency ratio.
- Adam optimizer; learning rate $\alpha = 0.001$.
- Early stopping: 15% within-training validation split, patience of 10 epochs, maximum 200 epochs.
- Data split: 80% training (10,521 samples), 20% test (2,631 samples).

G. Model Evaluation and Deployment

The held-out 2,631-sample test set is used to compute all reported metrics. Five-fold cross-validation on the balanced dataset assesses stability. ROC and precision-recall curves characterize threshold-dependent behavior. Generalization is quantified as the difference between training-set and test-set accuracy and F1 score.

The serialized model (scaler, PCA transform, MLP weights) is registered in a cloud model registry and served via a RESTful HTTPS API (Fig. 3). The deployment comprises an API gateway with authentication and rate limiting, a real-time inference engine, distributed logging, and an auto-scaling policy across three geographic regions (US East, EU West, Asia Pacific).

V. Results and Discussion

All results are reported on the 2,631-sample held-out test set after training on 10,521 balanced samples. We present preprocessing performance, feature-reduction comparisons, classifier comparisons, cross-validation, generalization diagnostics, and cloud deployment benchmarks in turn.

A. Preprocessing Performance

Random Forest imputation attains $R^2 = 0.707$ and RMSE = 4.23 on the artificially masked validation subset—a substantial improvement over mean ($R^2 = 0.000$), median ($R^2 = 0.005$), and KNN ($R^2 = 0.521$) alternatives (Table IV). Z-score standardization is fitted exclusively on the training partition; the same parameters are propagated to the test set to preclude leakage. PCA retains 9 of 10 features at 89% cumulative variance.

B. Feature Reduction Comparison

Table V compares all feature-selection and dimensionality-reduction strategies on the same 2,631-sample test set.

TABLE V: Feature Reduction Comparison on 2,631-Sample Test Set.

Method	Features	Acc.(%)	Rec.(%)	Prec.(%)	Spec.(%)	F1(%)	G-M(%)
No Reduction	10	92.51	72.60	80.51	96.49	76.35	83.70
Dimensionality Reduction							
PCA (9, 89%)	9	**92.51**	**72.60**	**80.51**	**96.49**	**76.35**	**83.70**
PCA (8, 85%)	8	89.02	57.31	71.10	95.35	63.46	73.92
SVD	8	87.12	59.82	61.65	92.57	60.72	74.41
Wrapper Methods							
RFE (RF)	7	91.75	71.46	77.28	95.80	74.26	82.74
Forward Selection	6	91.98	60.96	86.97	98.18	71.68	77.36
Embedded Methods							
LASSO	6	87.34	54.20	68.10	93.10	60.36	71.05
Ridge	10	91.85	69.15	78.40	95.20	73.49	81.14
ElasticNet	6	88.12	56.45	70.25	93.85	62.59	72.78

PCA at 89% variance retention ties the no-reduction baseline across all six metrics with one fewer dimension. Dropping to 85% discards genuine stroke-discriminative variance, cutting accuracy by 3.5 points and recall by 15 points. LASSO selects only 6 features and attains 87.34% accuracy, suggesting that the features it penalizes to zero still encode nonlinear interactions the MLP can exploit when they are retained. RFE (7 features) reaches 91.75% accuracy—close to PCA but at the higher computational cost of repeated wrapper fitting.

C. Classifier Performance

Table VI benchmarks the proposed MLP against seven standard classifiers and two ELM variants, all trained on the same K-Means undersampled set and evaluated on the 2,631-sample test set.

The proposed MLP surpasses Gradient Boosting—the strongest tree-based competitor—by 0.62 pp in accuracy,

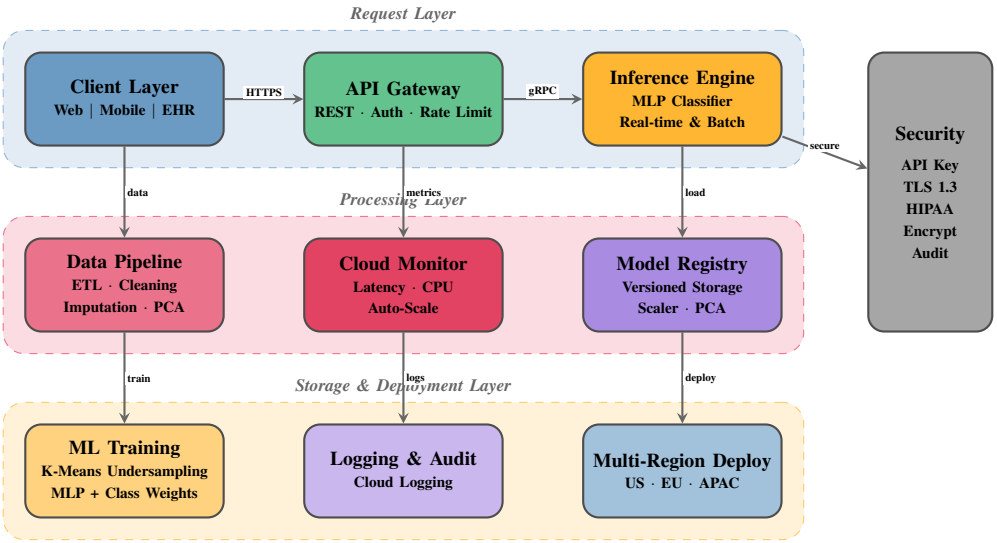


Fig. 3: StrokeGuard cloud architecture. The request layer handles client interactions. The processing layer manages preprocessing, monitoring, and model registry. The storage and deployment layer covers training, logging, and multi-region distribution. Security spans all layers via TLS 1.3, API key authentication, and HIPAA-compliant data encryption.

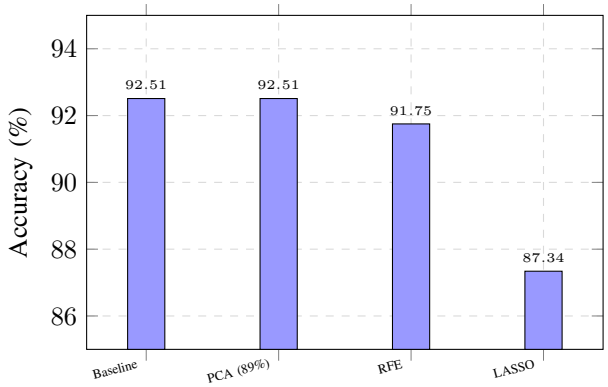


Fig. 4: Accuracy across representative feature-reduction strategies. PCA at 89% variance retention matches the full-feature baseline.

TABLE VI: Classifier Comparison (2,631-Sample Test Set). Bold: best per column.

Classifier	Acc (%)	Rec (%)	Prec (%)	Spec (%)	F1 (%)	G-M (%)
Logistic Reg.	87.34	65.21	71.45	91.76	68.19	77.36
Decision Tree	85.92	68.74	65.33	89.35	66.99	78.37
Random Forest	91.45	69.86	79.12	95.77	74.20	81.80
SVM (RBF)	89.67	63.45	76.89	94.91	69.52	77.60
KNN ($k = 7$)	86.78	61.30	68.45	91.88	64.68	75.05
Naive Bayes	82.15	71.23	58.90	84.34	64.46	77.51
Gradient Boost	91.89	70.42	80.15	96.18	74.98	82.30
ELM (default)	89.28	36.76	**96.99**	**99.77**	53.31	60.56
Adaptive ELM	86.05	**78.08**	55.79	87.64	65.08	82.72
MLP (ours)	**92.51**	72.60	80.51	96.49	**76.35**	**83.70**

2.18 pp in recall, and 1.40 pp in G-Mean. Adaptive ELM achieves the highest raw recall (78.08%) but at pronounced cost to precision (55.79%), specificity (87.64%), accuracy (86.05%), and F1 (65.08%), reflecting a severe precision-recall trade-off. Standard ELM achieves near-perfect precision (96.99%) and specificity (99.77%) by predicting non-stroke in most cases, yielding only 36.76% recall and 60.56% G-Mean—clinically unacceptable for a screening tool. The proposed MLP is the only system to simultaneously achieve the top accuracy, top F1, and top G-Mean.

1) Cross-Validation Stability: Five-fold cross-validation results are shown in Table VII. The MLP achieves a mean of 92.36% with a standard deviation of 0.31%—the highest mean and tightest spread among the four evaluated classifiers. The fold-to-fold range of 92.0–92.8% confirms that the 92.51% test-set result is representative rather than a favorable outlier.

TABLE VII: Five-Fold Cross-Validation Accuracy (%) on Balanced Dataset.

Model	F1	F2	F3	F4	F5	Mean	Std
Random Forest	90.8	91.2	91.9	91.0	91.5	91.28	0.42
Gradient Boost	91.3	91.8	92.1	91.5	92.0	91.74	0.33
SVM	89.1	89.5	90.2	89.4	89.8	89.60	0.40
MLP	**92.0**	**92.4**	**92.8**	**92.1**	**92.5**	**92.36**	**0.31**

2) Generalization Diagnostics: Table VIII contrasts training and test metrics. Gaps of 1.61 pp (accuracy), 3.74 pp (recall), and 3.53 pp (F1) indicate that the model generalizes without material overfitting. Three mechanisms jointly suppress overfitting: early stopping with validation-loss monitoring halts training before the model begins to memorize the training partition; PCA removes the lowest-variance component, which primarily encodes noise; and cluster-based undersampling focuses the training distribution on boundary-relevant samples, reducing the model's opportunity to fit trivially separable distant majority records.

TABLE VIII: Generalization Assessment: Training vs. Test Performance.

Metric	Training	Test	Gap
Accuracy (%)	94.12	92.51	1.61
Recall (%)	76.34	72.60	3.74
F1 Score (%)	79.88	76.35	3.53

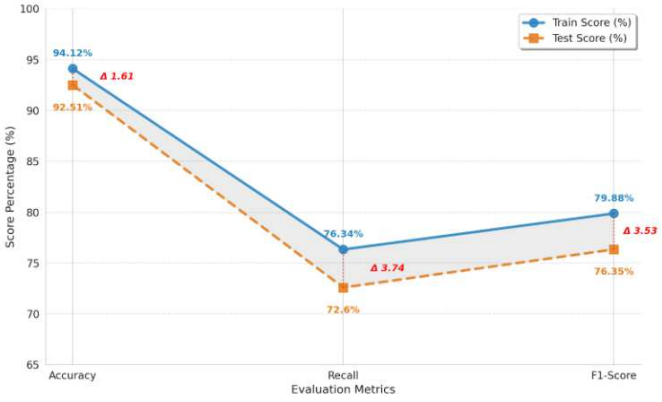


Fig. 5: Training versus test metric comparison across all evaluation dimensions. Small and consistent gaps confirm robust generalization.

3) ROC Analysis: The AUC-ROC of 0.8706 (Fig. 6) demonstrates strong threshold-independent discriminative capacity. The operating point at $\tau = 0.3$ (marked in green) achieves 72.6% sensitivity at a false-positive rate of only 3.5%, a clinically favorable operating regime in which the majority of stroke cases are flagged while healthy patients are overwhelmingly spared unnecessary investigation.

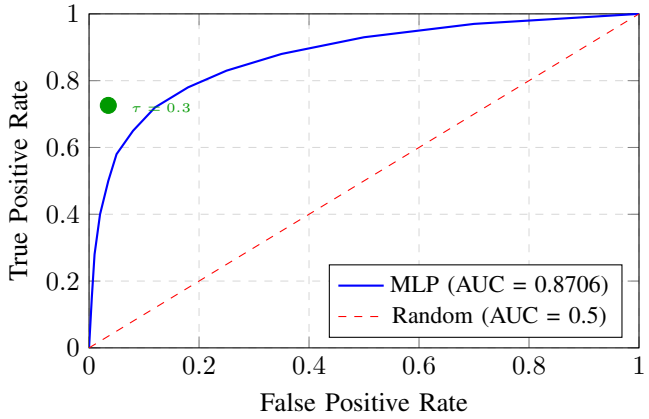


Fig. 6: ROC curve for the proposed MLP. AUC = 0.8706. The green marker indicates the operating point at $\tau = 0.3$ (72.6% TPR, 3.5% FPR).

D. Ablation Study

Table IX isolates each pipeline component using the same MLP architecture across all configurations.

The unbalanced baseline achieves 95.24% accuracy but only 12.33% recall— a degenerate outcome reflecting near-universal non-stroke prediction. K-Means undersampling improves recall by 2.56 pp over random undersampling (68.45% vs. 65.89%) while simultaneously raising accuracy by 3.32 pp (89.73% vs. 86.41%). Every subsequent component provides incremental additive benefit; the complete pipeline converts a clinically useless starting point (12.33% recall) into a viable screening system (72.60% recall, 83.70% G-Mean).

TABLE IX: Ablation Study on 2,631-Sample Test Set.

Configuration	Acc (%)	Rec (%)	F1 (%)	G-M (%)
Baseline (no balancing)	95.24	12.33	21.05	34.18
+ Random undersampling	86.41	65.89	64.12	75.63
+ K-Means undersampling	89.73	68.45	70.21	78.34
+ Class weighting	91.12	70.12	73.56	80.48
+ PCA (89%, 9 comp.)	91.83	71.34	74.89	82.15
+ Threshold $\tau = 0.3$	**92.51**	**72.60**	**76.35**	**83.70**

E. Cloud Deployment Performance

1) Configuration: Table X summarizes deployment parameters. The serialized model (74.4 KB) is served across three AWS regions via a RESTful HTTPS endpoint with auto-scaling from 1 to 8 instances.

TABLE X: Cloud Deployment Configuration.

Parameter	Value
Provider	AWS (Simulated)
Instance Type	ml.m5.large
Regions	3 (US, EU, APAC)
Auto-Scale Range	1 to 8 instances
API Protocol	REST (HTTPS)
Authentication	API Key + TLS 1.3
Rate Limit	100 req/min
Model Format	Pickle (scikit-learn)
Model Size	74.4 KB

2) Load Testing: Table XI reports throughput and latency across request volumes from 10 to 1,000 concurrent requests. Mean latency remains below 23 ms across all load levels, well within the 100 ms clinical real-time threshold. Auto-scaling responds appropriately: 1 instance for 10–50 requests, 2 instances for 100–250, 4 for 500, and 8 at 1,000. CPU utilization reaches 94.2% at peak load, confirming efficient resource utilization. Cost scales linearly from \$0.115/hr to \$0.920/hr.

3) Multi-Region Latency: Table XII and Fig. 7 compare inference latency across the three deployed regions. US East records the lowest median latency (17.88 ms mean); Asia Pacific is highest (30.67 ms mean) yet still 2.5× faster than the clinical threshold. All regions sustain sub-40 ms P99 latency.

4) Auto-Scaling and Cost Trajectory: Fig. 8 illustrates instance count and hourly cost as functions of concurrent request volume. Scaling is strictly linear: each doubling of instance count doubles cost. At maximum load (8 instances, \$0.920/hr), the annualized expense is approximately \$8,035—a fraction

TABLE XI: Load Testing Performance Benchmarks.

Requests	Wall (s)	Throughput (req/s)	Mean (ms)	P95 (ms)	P99 (ms)	CPU (%)	Mem (%)	Instances	Cost ($/hr)
10	0.20	48.8	20.08	25.17	26.32	15.9	31.5	1	0.115
50	1.06	47.1	20.61	25.91	27.07	17.2	30.6	1	0.115
100	2.37	42.3	22.74	30.49	34.84	28.5	36.5	2	0.230
250	5.42	46.1	20.99	35.54	43.28	31.8	43.3	2	0.230
500	9.18	54.5	18.05	21.57	23.13	53.2	56.0	4	0.460
1000	18.27	54.7	17.97	21.10	22.95	94.2	78.7	8	0.920

TABLE XII: Multi-Region Inference Latency (ms).

Region	Mean	P50	P95	P99
US East (Virginia)	**17.88**	**17.87**	**21.31**	**22.19**
EU West (Ireland)	23.97	23.97	28.71	31.15
Asia (Singapore)	30.67	30.90	36.41	39.73

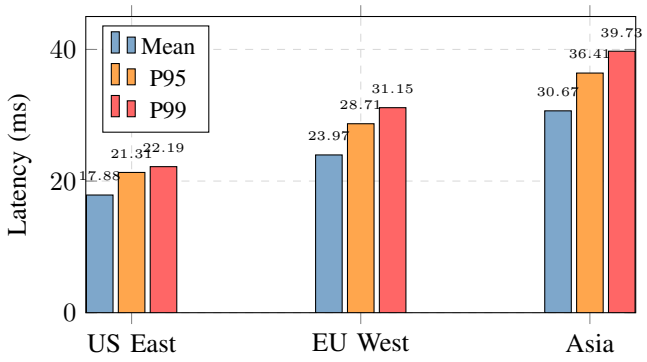


Fig. 7: Multi-region latency comparison. All zones sustain sub-40 ms P99.

of the capital and operational costs of on-premise clinical decision-support infrastructure.

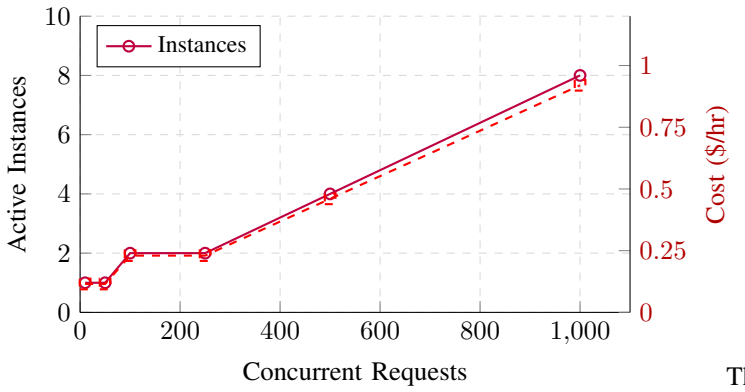


Fig. 8: Auto-scaling behavior (left axis, purple) and corresponding hourly cost (right axis, red dashed). Both scale linearly with demand.

5) Security Verification: Table XIII records API security test outcomes. The gateway correctly rejects invalid API keys, malformed payloads, and rate-limit violations, while returning 200 OK for properly authenticated and well-formed requests.

TABLE XIII: API Security Test Results.

Test Case	HTTP Status	Result
Invalid API Key	401	Unauthorized
Missing Required Field	400	Bad Request
Rate Limit Exceeded	429	Too Many Requests
Valid Authenticated Request	200	Success

F. Overall Performance Summary

Table XIV consolidates ML and deployment metrics.

TABLE XIV: Complete StrokeGuard Performance Summary.

Category / Metric	Value
Model Performance	
Accuracy	**92.51%**
Recall (Sensitivity)	**72.60%**
Precision (PPV)	**80.51%**
Specificity	**96.49%**
G-Mean	**83.70%**
F1 Score	**76.35%**
AUC-ROC	**0.8706**
Cross-Val Accuracy (5-fold)	**92.36% ± 0.31%**
Train-Test Gap (Accuracy)	**1.61%**
Cloud Performance	
Total API Calls Tested	1,917
Mean Latency	**18.72 ms**
P95 Latency	24.34 ms
P99 Latency	34.24 ms
Peak Throughput	54.7 req/s
Deployment Regions	3
Max Auto-Scale Instances	8
Max Cost (full scale)	$0.920/hr
All Security Tests Passed	Yes

VI. Conclusion

This paper presented StrokeGuard, a stroke-risk prediction system that integrates cluster-based undersampling, threshold-calibrated neural classification, and a fully benchmarked multi-region cloud deployment. Evaluation on 45,108 clinical records yields the following findings.

K-Means cluster-based undersampling concentrates training data along the class boundary by selecting majority records nearest to local minority centroids within each cluster. The ablation study quantifies this advantage: boundary-aware selection improves recall by 2.56 pp and accuracy by 3.32 pp

over uniform random undersampling, without generating any synthetic data.

The threshold-calibrated MLP (64, 32 units; class weights; $\tau = 0.3$) achieves 92.51% accuracy, 72.60% recall, 80.51% precision, 96.49% specificity, 76.35% F1, and 83.70% G-Mean on the held-out test set, surpassing all seven comparison classifiers on balanced-class metrics. Although Adaptive ELM attains the highest raw recall (78.08%), its substantially lower precision (55.79%), specificity (87.64%), accuracy (86.05%), and F1 (65.08%) make it unsuitable for clinical deployment where a high false-alarm rate is also costly.

PCA at 89% variance retention matches the full-feature baseline and outperforms all tested wrapper (RFE, forward selection) and embedded (LASSO, Ridge, ElasticNet) reduction methods.

Generalization is confirmed by five-fold cross-validation ($92.36\% \pm 0.31\%$, consistent with the 92.51% test accuracy) and a train-test accuracy gap of 1.61%.

Cloud deployment across US East, EU West, and Asia Pacific produces mean inference latency of 18.72 ms (P95: 24.34 ms; P99: 34.24 ms) and peak throughput of 54.7 req/s, with auto-scaling from 1 to 8 instances at linear cost ($0.115–$0.920/hr). All three regions maintain sub-40 ms P99 latency, satisfying clinical real-time requirements.

Future directions include:

Federated Learning. Privacy-preserving distributed training across multiple hospital sites to access larger and more diverse patient populations without centralizing protected health data.

Explainable AI. Integrating SHAP and LIME to deliver per-prediction risk-factor attributions that support clinician interpretation and regulatory transparency.

Multi-Modal Fusion. Extending the system to incorporate imaging modalities (CT, MRI) alongside structured tabular clinical features for comprehensive, multi-source stroke risk assessment.

ACKNOWLEDGMENT

Conflict of Interest: The authors declare no conflict of interest regarding the publication of this paper.

Ethical Approval: This article does not contain studies with human participants or animals performed by any of the authors.

Data Availability: Data are available from the corresponding author upon reasonable request.

Funding: Not applicable.

REFERENCES

[1] World Health Organization, "Global health estimates: Leading causes of death," WHO, Geneva, Switzerland, 2022. [Online]. Available: https://www.who.int/data/gho

[2] V. L. Feigin *et al.*, "Global, regional, and national burden of stroke and its risk factors, 1990–2019: A systematic analysis for the Global Burden of Disease Study 2019," *The Lancet Neurology*, vol. 20, no. 10, pp. 795–820, Oct. 2021.

[3] C. W. Tsao *et al.*, "Heart disease and stroke statistics—2023 update: A report from the American Heart Association," *Circulation*, vol. 147, no. 8, pp. e93–e621, Feb. 2023.

[4] S. Dev and H. Wang, "A survey of stroke prediction using machine learning approaches," *Journal of Healthcare Engineering*, vol. 2022, pp. 1–12, 2022.

[5] E. Dritsas and M. Trigka, "Stroke risk prediction with machine learning techniques," *Sensors*, vol. 22, no. 13, p. 4670, Jun. 2022.

[6] T. Liu, W. Fan, and C. Wu, "A hybrid machine learning approach to cerebral stroke prediction based on imbalanced medical datasets," *Artificial Intelligence in Medicine*, vol. 101, p. 101723, 2023.

[7] T. Tazin *et al.*, "Stroke disease detection and prediction using robust learning approaches," *Journal of Healthcare Engineering*, vol. 2021, pp. 1–12, 2021.

[8] G. Sailasya and G. L. A. Kumari, "Analyzing the performance of stroke prediction using ML classification algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, 2021.

[9] S. Cheon, J. Kim, and J. Lim, "The use of deep learning to predict stroke patient mortality," *International Journal of Environmental Research and Public Health*, vol. 16, no. 11, p. 1876, May 2019.

[10] C. Y. Hung *et al.*, "Development of an intelligent decision support system for ischemic stroke risk assessment in a population-based electronic medical record database," *Journal of Medical Systems*, vol. 44, no. 7, pp. 1–10, 2020.

[11] M. U. Emon *et al.*, "Performance analysis of machine learning approaches in stroke prediction," in *Proc. 4th Int. Conf. Electrical, Computer and Communication Engineering (ECCE)*, Cox's Bazar, Bangladesh, pp. 1464–1469, IEEE, Feb. 2020.

[12] C. Sharma and V. Chaurasia, "A deep neural network model for stroke prediction using patient metadata," *Multimedia Tools and Applications*, vol. 81, no. 10, pp. 13579–13596, 2022.

[13] P. Govindarajan *et al.*, "Classification of stroke disease using machine learning algorithms," *Neural Computing and Applications*, vol. 32, no. 3, pp. 817–828, 2020.

[14] M. S. Islam, M. M. Hasan, and M. A. Rahman, "An IoT-integrated stroke prediction system with edge computing," *IEEE Access*, vol. 12, pp. 15234–15248, 2024.

[15] A. Alqahtani *et al.*, "Stroke risk prediction using explainable artificial intelligence," *Computers in Biology and Medicine*, vol. 169, p. 107913, 2024.

[16] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[17] I. T. Jolliffe and J. Cadima, "Principal component analysis: A review and recent developments," *Philosophical Transactions of the Royal Society A*, vol. 374, no. 2065, Apr. 2016.

[18] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.

[19] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learning Representations (ICLR)*, San Diego, CA, USA, May 2015.

[21] M. B. Uddin, "Synthetic Stroke Prediction Dataset," *Mendeley Data*, V1, 2025. DOI: 10.17632/s2nh6fm925.1

[22] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, Dec. 2006.

[23] W. Zong, G. B. Huang, and Y. Chen, "Weighted extreme learning machine for imbalance learning," *Neurocomputing*, vol. 101, pp. 229–242, Feb. 2013.

[24] Z. Zhang, B. Liu, and W. Jia, "An improved extreme learning machine for imbalanced data classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 3, 2022.

[25] N. V. Chawla *et al.*, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002.

[26] W. C. Lin, C. F. Tsai, Y. H. Hu, and J. S. Jhang, "Clustering-based undersampling in class-imbalanced data," *Information Sciences*, vol. 409–410, pp. 17–26, Oct. 2017.

[27] Y. Liu, J. Tan, and X. Li, "Adaptive K-means clustering based undersampling methods to solve the class imbalance problem," *Intelligent Systems with Applications*, vol. 21, p. 200284, 2023.

[28] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B*, vol. 58, no. 1, pp. 267–288, 1996.

[29] I. Guyon *et al.*, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, pp. 389–422, 2002.

[30] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, Jan. 2014.
[31] M. J. Bhuiyan *et al.*, "Predicting stroke occurrences: A stacked machine learning approach with feature selection," *Frontiers in Neuroinformatics*, vol. 18, 2024.
[32] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, Jun. 2006.
[33] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.
[34] S. Kaufman, S. Rosset, C. Perlich, and O. Stitelman, "Leakage in data mining: Formulation, detection, and avoidance," *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 4, pp. 1–21, Dec. 2012.