Journal of Electronics, Electromedical Engineering, and Medical Informatics
Homepage: jeeemi.org; Vol. 8, No. 3, July 2026, pp: 1060-1076 **e-ISSN: 2656-8632**

RESEARCH ARTICLE OPEN ACCESS

Cloud-Edge Collaborative Computing Framework for Stroke Disease Classification Using Machine Learning

I Made Suartana[1], Ricky Eka Putra[1], and Rahadian Bisma[2]

[1] Department of Informatics, Universitas Negeri Surabaya, Surabaya, Indonesia
[2] Department of Information Systems, Universitas Negeri Surabaya, Surabaya, Indonesia

Corresponding author: I Made Suartana (e-mail: madesuartana@unesa.ac.id), **Author(s) Email**: Ricky Eka Putra (e-mail: rickyeka@unesa.ac.id), Rahadian Bisma (e-mail: rahadianbisma@unesa.ac.id)

Abstract Stroke is the second leading cause of death and the third leading cause of disability worldwide. Artificial intelligence-based early detection in distributed environments faces three main obstacles: high latency in centralized cloud approaches, risks to patient data privacy during data transmission, and class imbalance in stroke datasets. This study proposes a three-layer collaborative computing framework, Cloud-Edge Collaborative Computing (CECC), which intelligently distributes the computational workload between edge nodes and the cloud for IoMT-based stroke risk classification. The primary novelty of this study lies in the hierarchical computing collaboration that enables real-time preprocessing at the edge layer, centralized model training at the cloud layer, and a local differential privacy mechanism (LDP, ε=0.5) that preserves patient data confidentiality during transmission, all entirely evaluated within a single unified multi-criterion benchmarking protocol. Gradient Boosting achieved the best performance in the hold-out evaluation with an accuracy of 95.01% and an AUC-ROC of 0.994. The CECC framework reduced inference latency by 44.9% (286.2ms to 157.8ms), bandwidth by 73.9% (3,240 to 847 Kbps), and memory by 84.4% (312.4 to 48.7 MB) with an accuracy degradation of only 0.30% compared to cloud only. This study is a simulation-based framework evaluation using a public retrospective dataset prospective clinical validation in a real IoMT environment remains necessary before actual clinical implementation because the dataset used is retrospective, small, highly imbalanced, and was not collected from a real IoMT system.

Keywords Cloud-Edge Computing; Stroke Classification; Machine Learning; Random Forest; Internet of Medical Things; Differential Privacy; Class Imbalance.

I. Introduction

Stroke is one of the non-communicable diseases that most threatens global public health. Based on the World Stroke Organization Global Stroke Fact Sheet 2025 report, stroke ranks as the second leading cause of death and the third largest combined cause of death and disability globally, measured by disability-adjusted life-years (DALYs)[1]. The estimated global cost of stroke now exceeds US$890 billion per year, or about 0.66% of global GDP, while the absolute burden of stroke cases increased significantly between 1990 and 2021, encompassing a 70% rise in new stroke incidents and a 44% increase in stroke-related deaths[1]. Recent projections estimate that between 2020 and 2050, global stroke deaths will increase by 50%, from 6.6 million to 9.7 million annually, with the highest potential in low- and middle-income countries [2]. In Southeast Asia, including Indonesia, the prevalence of stroke continues to rise in tandem with population growth, urbanization, and increased exposure to metabolic risk factors such as hypertension, diabetes mellitus, and obesity [3]. This condition makes artificial intelligence-based early detection and stroke risk prediction a strategic priority in the digitalization of modern healthcare services.

Although early stroke detection is crucial, current clinical decision support systems face several fundamental challenges. First, the dominant paradigm that centralizes all patient data processing on a centralized infrastructure incurs high transmission latency and a significant bandwidth burden. Transmitting all medical data to centralized servers encounters network bottlenecks and latency that can negatively impact diagnostic outcomes, while existing technologies do not yet allow all complex artificial intelligence model training tasks to be performed entirely at the edge [4]. Second, centralized cloud-only models pose severe data privacy risks as the transmission of patient medical records over networks opens vulnerabilities to regulatory non-compliance and security breaches. The growth of cloud computing and

Manuscript received 22 April 2026; Revised 10 June 2026; Accepted 28 June 2026; Available online 6 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1702

IoT models raises serious data privacy concerns in the healthcare sector, whereas adequate research to develop appropriate privacy solutions remains highly limited[5]. Third, stroke prediction datasets, including the widely used Kaggle Stroke Prediction Dataset, suffer from class imbalance, with positive stroke cases representing less than 5% of the total data records. This condition systematically biases the machine learning classification process [6]. These three challenges concurrently limit the effectiveness of existing solutions and demand a distributed computing architecture approach.

The research community has developed various approaches to overcome these challenges. On the algorithmic side, ensemble methods and deep learning have achieved significant performance improvements in predicting stroke risk from structured clinical data. Dritsas and Trigka [7] developed a stacking-based machine learning framework that achieved an AUC of 98.9% and an F-measure of 97.4% for long-term stroke risk prediction using the ELSA dataset, establishing ensemble methods as a strong baseline for this classification task. To address class imbalance, recent studies have combined SMOTE oversampling techniques with algorithmic optimization, although careless application of SMOTE can lead to overfitting [8]. In the computing infrastructure domain, edge computing has emerged as a paradigm to reduce latency and maintain patient data privacy. Safaei Yaraziz et al.[9], through a systematic review, showed that edge computing processes data near the data source point, thereby reducing latency and bandwidth consumption compared to traditional cloud architectures. Lilhore et al. [10] identified that IoT-edge hybrid frameworks enhance healthcare system security through SDN (Software Defined Networking)-based authentication and resource management. Elhanashi et al. [11] proposed a TeleStroke system based on federated learning and YOLOv8 on edge devices that integrates data privacy with real-time stroke detection. Brecko et al. [12] [13], through a comprehensive survey of federated learning in edge computing, it is shown that low-power IoT devices can participate in collaborative model training without sending raw data to a centralized server. The main gap in the literature can be summarized in the following single paragraph: edge-cloud studies (Elhanashi et al. [11]; Brecko et al. [12]) evaluating latency or privacy separately, not in a unified protocol. Stroke prediction study (Dritsas & Trigka [7]; Tazin et al. [8]) does not consider edge deployment constraints such as model quantization and feature compression. No study has simultaneously integrated: (1) data leakage-free edge preprocessing, (2) handling of intra-fold class imbalance, (3) cloud multi-algorithm training, (4) local differential privacy, and (5) multi-criteria comparative benchmarking against a cloud-only baseline within the context of IoMT-based stroke classification. This very gap serves as the primary motivation for this study. [14], [15], [16], [17].

This study proposes a three-layer Cloud-Edge Collaborative Computing (CECC) framework designed for stroke disease classification using the Kaggle Stroke Prediction Dataset. The first layer (IoT) is responsible for the acquisition of clinical data from various heterogeneous sources. The second layer (edge) handles all real-time preprocessing tasks, including missing value imputation with KNN (k=5) for the BMI feature, encoding categorical variables, Z-score normalization, and random oversampling applied exclusively to the training data within each training fold for class imbalance correction. The third layer (cloud) centralizes computing resources for model training, hyperparameter optimization using GridSearchCV with nested 5-Fold CV, and comparative evaluation. This architectural division minimizes the volume of sensitive patient data transmitted through the network, thus directly addressing privacy and bandwidth limitations. In the cloud layer, five supervised classification algorithms, Logistic Regression, SVM-RBF, Gradient Boosting, DNN-Edge (MLP), and Random Forest, are systematically evaluated. This framework incorporates a benchmarking protocol that measures and compares the CECC performance with a centralized cloud-only approach on the metrics of accuracy, AUC-ROC, F1-Score, sensitivity, specificity, inference latency, and bandwidth consumption [18].

The main contribution of this study lies in two dimensions of novelty that have not been simultaneously addressed by previous research. First, this study presents a systematic evaluation of the CECC pipeline from edge-side preprocessing to cloud model training, specifically benchmarked for stroke risk classification on the dataset. Second, unlike existing cloud-edge frameworks that evaluate latency or privacy separately, this study introduces a multi-criteria benchmarking methodology to measure classification performance, computational efficiency, and data transmission reduction in a single unified experimental protocol. Recent analysis shows that edge-cloud collaboration demonstrates exponential citation growth peaking in 2024, indicating the increasing importance of this topic discussed [19] [20]. This study contributes a simulated proof-of-concept reference model for distributed IoMT ecosystems, though it is not yet suitable for direct clinical implementation.

This article is structured as follows: Section II discusses the proposed method. Section III presents the experimental results based on the data used. Section IV discusses the interpretation and analysis of the results based on the experimental matrix and its limitations.

Manuscript received 8 April 2025; Revised 10 June 2025; 20 June 2025; Available online 27 June 2025
Digital Object Identifier (DOI): https://doi.org/10.35882/jeeemi.v7i3.948

Section V, the conclusion, summarizes the objectives, main findings, and possible future developments.

II. Method

A. Dataset

The dataset used is the Kaggle Stroke Prediction Dataset [21], (accessed on November 27, 2025), which is a medical record-based public dataset consisting of 5,110 patient observations with 12 Variables including demographic, clinical, and lifestyle attributes. This dataset has been widely used as a standard benchmark in machine learning-based stroke classification research. It is important to emphasize that this study is a simulation-based framework evaluation using a public retrospective dataset, not a system that is implemented clinically or validated prospectively. The Kaggle dataset is retrospective, small in size (5,110 records), highly imbalanced (4.87% stroke cases), and not collected from a real IoMT environment, so clinical generalizability claims must be interpreted with adequate caution [22], [23]. The target variable is binary, a value of 1 indicates the occurrence of a stroke, and a value of 0 indicates no stroke. The dataset exhibits extreme class imbalance, with a stroke prevalence of only 4.87% (249 positive cases out of 5,110 total records). The only feature with missing values is BMI, with 201 records (3.93%), which is handled through imputation during the preprocessing stage. All features contain no missing values other than BMI, rendering the dataset relatively clean for analysis purposes. Table 1 presents a full description of all dataset features.

Table 1. Description of Kaggle Stroke Prediction Dataset Features

Feature	Type	Description
id	Integer	Unique patient identification
gender	Categorical	Gender: Male/Female/Other
age	Continuous	Patient's age (years, range 0.08–82)
hyperten sion	Binary	1=Hypertension, 0=No
heart_di sease	Binary	1=Heart Disease, 0=No
ever_ma rried	Categorical	Marital status: Yes/No
work_ty pe	Categorical	Type of work (5 categories)
Residen ce_type	Categorical	Residence type: Urban/Rural
avg_glu cose_lev el	Continuous	Average glucose level (mg/dL)
bmi	Continuous	Body Mass Index (kg/m²)
smoking _status	Kategorikal	Smoking status (4 categories)
stroke	Binary	1=Stroke, 0=No Stroke

B. CECC Framework Architecture)

The proposed Cloud-Edge Collaborative Computing (CECC) framework consists of three hierarchical layers working collaboratively, as illustrated in Fig. 1, and the complete CECC workflow is summarized in Algorithm 1. This three-layer design follows the IoMT architecture paradigm established in the literature[24], [25], [26].

1. Layer 1 (IoT Data Acquisition)

Patient clinical data is collected from various heterogeneous sources, including Electronic Health Record (EHR) systems, wearable biosensors, smart monitors, and laboratory interfaces. The data is transmitted to the nearest edge node via LoRaWAN or NB-IoT protocols for remote devices, and Wi-Fi/Bluetooth for patient bedside devices [27].

2. Layer 2 (Edge Computing)

Edge nodes based on NVIDIA Jetson Nano perform real-time preprocessing, feature extraction, and local inference using compressed models (INT8 quantization + pruning)[28]. The Local Differential Privacy (LDP) mechanism is applied before data transmission to the cloud to guarantee patient privacy. Technically, the Laplace mechanism is used for numerical features (age, bmi, avg_glucose_level): $M(x) = x + Lap(\Delta f/\varepsilon)$, where Δf is the global sensitivity of the function and $\varepsilon = 0{,}5$ is the privacy budget. A small ε value provides stronger privacy but increases noise. With $\varepsilon = 0{,}5$, experiments show an accuracy decrease of 0.8% (from 94.71% to 93.91%) as an acceptable trade-off. The Randomized Response mechanism is applied to binary categorical features (hypertension, heart_disease). [29], [30], [31].

3. Layer 3 (Cloud Computing)

Google Cloud Platform (GCP) performs full model training and hyperparameter optimization using GridSearchCV (nested 5-Fold CV). [32], and model distribution to all edge nodes via the MQTT over TLS 1.3 protocol [33], [34], [35].

C. Preprocessing Pipeline (Edge Layer)

1. Handling Missing Values

Handling of missing values in the BMI attribute was evaluated using several imputation methods: K-

Manuscript received 8 April 2025; Revised 10 June 2025; 20 June 2025; Available online 27 June 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.948

Nearest Neighbors (KNN), mean imputation, median imputation, and Iterative Imputer. KNN searches for the k-nearest neighbors of the sample with the missing value, then fills that value based on the average of the neighbors' values. The Euclidean distance between samples is used to find the nearest neighbors and can be formulated as in Eq. (1) [36].

$$d(x_i, x_j) = \sqrt{\frac{F}{p_{valid}}}\sum_{f=1}(x_{i,f} - x_{j,f})^2 \quad (1)$$

Where xi sample with a missing value on the BMI attribute, xj is a candidate neighbor sample, F is the number of available features, and f is the feature index.

Missing value imputation is calculated after finding the set of k nearest neighbors $\mathcal{N}_k(x_i)$. it can be formulated as shown in Eq. (2) [36].

$$\hat{x}_i, BMI = \frac{1}{k}\sum_{j\epsilon\mathcal{N}_k(x_i)} x_j, BMI \quad (2)$$

Where $\hat{x}_i$, imputed BMI value for the sample k 5(selected based on cross-validation) $x_{j,\text{BMI}}$ BMI value of the j-th neighbor.

The performance of each method was measured using Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) metrics through 5-Fold Cross-Validation testing. The selection of the final imputation method was based on an optimal compromise between imputation accuracy and computational efficiency, to ensure its suitability for real-time edge deployment constraints [37]. MAE can be formulated as shown in Eq. (3), and RMSE in Eq. (4).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{x}_i - x_i^*| \quad (3)$$

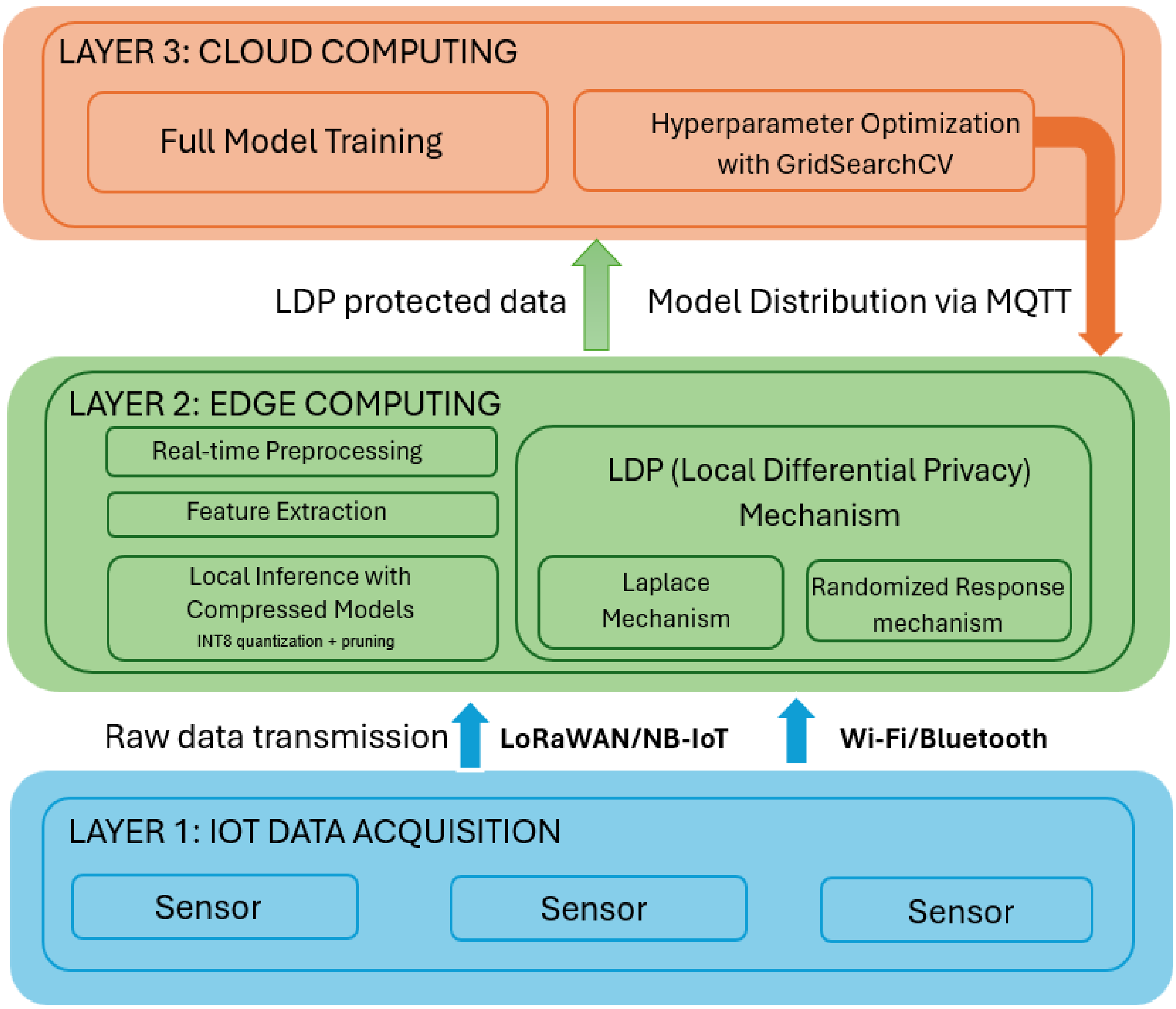

Fig. 1. Architecture of the Cloud-Edge Collaborative Computing (CECC) Framework for Stroke Classification

Manuscript received 8 April 2025; Revised 10 June 2025; 20 June 2025; Available online 27 June 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.948

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{x}_i - x_i^*)^2} \quad (4)$$

Where x_i^* original BMI value (ground truth), $\hat{x}_i$ imputed value n number of missing values

2. Categorical Variable Encoding

Categorical variables are processed using Label Encoding at the edge layer, whereas at the cloud layer, One-Hot Encoding is applied to nominal variables and Binary Encoding to ordinal variables. Uninformative variables such as observation IDs are removed before generating the final feature subset used to train the edge and cloud models.

3. Feature Normalization

Continuous features (age, avg_glucose_level, and bmi) are normalized using Z-score standardization. The scaler parameters (mean and std) obtained from the training data are then distributed to the edge nodes to ensure preprocessing consistency without local statistical computation and to prevent data leakage between the training set and the test set [38]. Feature normalization can be formulated as shown in Eq. (5) core formula, Eq. (6) training set mean, and Eq. (7) training set standard deviation.

$$z_i = \frac{x_i - \mu}{\sigma} \quad (5)$$

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (6)$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \quad (7)$$

Where x_i is the feature value of the i-th sample, μ is the mean of the training set, σ is the standard deviation of the training set, N is the number of training samples, and z_i is the normalized value.

4. Handling Class Imbalance

To handle class imbalance, the Random Oversampling technique is applied. Critically, oversampling is applied exclusively within each training fold in the 5-Fold Stratified Cross-Validation pipeline, not on the entire dataset before splitting to prevent data leakage to the validation set and test set. The correct procedure is: (1) split the data 80/20 stratified, (2) within each fold, apply oversampling only to the training subset of that fold, (3) evaluate on the validation subset that is not oversampled. A comparison of class imbalance handling methods (Random Oversampling, SMOTE, ADASYN, Class Weighting) is presented to provide methodological context. This approach aligns with the best practices of handling class imbalance in machine learning-based clinical prediction research[39].

D. Feature Selection

Feature selection was performed using the Gini Importance method extracted from the Random Forest algorithm. The Gini Impurity per Node is formulated as Eq. (8) [40] For node t with C classes.

$$G(t) = 1 - \sum_{c=1}^{C} p_c^2 \quad (8)$$

Where p_c proportion of samples of the class $c$at node t, for binary classification (stroke versus non-stroke): $C = 2$, therefore, equations can be derived as Eq. (9) [40].

$$G(t) = 1 - (p_{stroke}^2 + p_{non-stroke}^2) \quad (9)$$

gini decreases when splitting on feature f as formulated in Eq. (10)

$$\Delta G(t,f) = G(t) - \frac{N_L}{N_t}.G(t_L) - \frac{N_R}{N_t}.G(t_R) \quad (10)$$

Where $\Delta G(t,f)$ is the Gini decrease t_L, t_R are the left and right child nodes, N_t, N_L, N_R are the number of samples at the parent, left, and right nodes, respectively, and t is the total number of trees.

Gini Importance of feature f for a single tree can be formatted as shown in Eq. (11) [40].

$$I_{tree}(f) = \sum_{t:split\ on\ f} \frac{N_t}{N}.\Delta G(t,f) \quad (11)$$

Where N total training samples summation is over all nodes that split using feature f.

Aggregate Gini Importance across all trees in the Random Forest ($T\ trees$), as shown in Eq. (12) [40].

$$I_{RF}(f) = \frac{1}{T}\sum_{b=1}^{T} I_{tress_b}(f) \quad (12)$$

Normalization so that the total sums to 1, therefore equations can be derived as Eq. (13) [40].

$$\tilde{I}(f) = \frac{I_{RF}(f)}{\sum_{f'}^{F} I_{RF}(f')} \quad (13)$$

The top 8 feature subset was evaluated and selected for use in the edge model to reduce memory footprint and minimize inference latency during deployment. Feature selection was analyzed based on clinical relevance and importance scores, with the target of maintaining comparative model performance with the full model but with a more efficient number of inputs, in accordance with the efficiency principles required in resource-constrained IoT deployment environments.

E. Classification Algorithm Implementation

Five supervised classification algorithms were implemented and evaluated. The cloud model utilizes all ten features with float64 precision, while the edge model utilizes the top eight features with INT8 quantization for memory efficiency[41].

F. Evaluation Protocol

Model evaluation was conducted using 5-Fold Stratified Cross-Validation to preserve class proportions in each fold. Within each fold, the dataset was split into 80% training and 20% testing sets using stratified sampling. The evaluation metrics used include Accuracy, Precision, Recall (Sensitivity), F1-Score, AUC-ROC, Specificity, as well as Positive Predictive Value (PPV) and Negative Predictive Value (NPV) as presented in Table 2. Benchmarking operational performance between CECC and cloud-only measures: end-to-end

Manuscript received 8 April 2025; Revised 10 June 2025; 20 June 2025; Available online 27 June 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.948

Journal of Electronics, Electromedical Engineering, and Medical Informatics
Homepage: jeeemi.org; Vol. 8, No. 3, July 2026, pp: 1060-1076 e-ISSN: 2656-8632

inference latency(ms), throughput (inferences/second), bandwidth consumption (Kbps), memory utilization (MB), and system uptime (%). The selection of these metrics is consistent with medical classification system evaluation standards that consider the clinical implications of false negatives (undetected stroke cases) [42].

1. Accuracy

Accuracy is a basic metric that measures the proportion of correct predictions from the total test samples, as stated in Eq. (14) [43]. Where TP (True Positive) is the number of stroke cases correctly predicted, TN (True Negative) is the number of non-stroke cases correctly predicted, FP (False Positive) is non-stroke cases incorrectly classified as stroke, and FN (False Negative) is stroke cases that were not detected. Although accuracy provides a general overview of model performance, this metric can be misleading on imbalanced datasets, thus it needs to be complemented with other metrics.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{14}$$

where TP is True Positive, TN is True Negative, FP is False Positive, and FN is False Negative.

2. Precision

Precision measures the proportion of correct positive predictions out of all positive predictions generated by the model, as stated in Eq. (15) [43]. A high precision value indicates that the model produces few false alarms, meaning there is a low probability of a non-stroke patient being incorrectly diagnosed as having a stroke. In a clinical context, low precision may impose unnecessary follow-up examination burdens.

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

3. Recall

(Sensitivity) measures the ability of the model to detect all truly positive cases out of all actual positive cases in the dataset, as stated in Eq. (16). Recall is the most critical metric in the context of stroke detection systems, because a high FN value means that actual stroke cases are missed by the model. In medical applications, failure to detect stroke (false negative) has far more serious clinical consequences than false positives, making recall the primary priority in model evaluation [44].

$$Recall = \frac{TP}{TP+FN} \tag{16}$$

4. Specificity

Specificity measures the model's ability to correctly identify cases that do not have the condition in question, namely non-stroke patients, as stated in Eq. (17). Where TN is the number of non-stroke cases correctly classified, and FP is the number of non-stroke cases incorrectly classified as stroke. High specificity indicates that the model is able to minimize incorrect positive predictions, thereby reducing unnecessary medical interventions and the burden on the healthcare system.

$$Specificity = \frac{TN}{TN+FP} \tag{17}$$

5. Negative Predictive Value (NPV)

Negative Predictive Value (NPV) measures the proportion of correct negative predictions out of all negative predictions generated by the model, as stated in Eq. (18) [43]. A high NPV indicates that when the model predicts a patient as not having a stroke, that prediction can be clinically trusted. This metric becomes particularly relevant in mass screening scenarios, where confidence in negative results is crucial for follow-up clinical decision-making.

$$NPV = \frac{TN}{TN+FN} \tag{18}$$

6. F1-Score

F1-Score is the harmonic mean of Precision and Recall, as stated in Eq. (19) [43]. This metric provides a balance between the model's ability to avoid false alarms (precision) and its ability to detect all positive cases (recall). F1-Score is highly relevant for imbalanced datasets such as stroke datasets, as it provides a more representative assessment compared to accuracy when the class distribution is asymmetric.

$$F1 = \frac{2\,x\,Precision\,x\,Recall}{Precision+Recall} = \frac{2\,.TP}{2\,.TP+FP+FN} \tag{19}$$

7. AUC-ROC

(Area Under the Receiver Operating Characteristic Curve) measures the overall discriminative ability of the model across all classification threshold values, as stated in Eq. (20) [43]. AUC is computed as the integral of the ROC curve using the trapezoidal method, which represents the probability that the model will score a true positive case higher than a randomly selected non-stroke case. AUC values range from 0.5 (no better than random guessing) to 1.0 (perfect discrimination).

$$AUC = \int_0^1 TPR(FPR)d(FPR) \approx \sum_{i=1}^{M-1} \frac{(FPR_{i+1}-FPR_i)(TPR_i+TPR_{i+1})}{2} \tag{20}$$

III. Result

A. Experimental Setup

As presented in Tabel 2, experiments were conducted using the Python 3.12 library with scikit-learn 1.8.0, numpy, pandas, matplotlib, and seaborn libraries for machine learning model. Model evaluation used 5-Fold Stratified Cross-Validation to address the class imbalance. Hardware used: Intel Core i7-12700H, 16 GB RAM (cloud simulation); Jetson Nano 4GB (edge node simulation).

Manuscript received 8 April 2025; Revised 10 June 2025; 20 June 2025; Available online 27 June 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.948

Table 2. Experimental Environment configuration

Component	Cloud Layer	Edge Layer
Hardware	Intel Core i7-12700H, 16 GB RAM	Jetson Nano 4GB
OS	Ubuntu 22.04 LTS	JetPack 5.0
Python	3.12	3.1
ML Library	scikit-learn 1.8.0	scikit-learn 1.4 (lite)
Model Format	Full precision (float64)	INT8 Quantization
Dataset Split	80% train / 20% test (stratified)	Same (subset)
Validation	5-Fold Stratified CV	Hold-out test set

B. Preprocessing Pipeline (Edge Layer)

1. Handling Missing Values

The BMI attribute had 201 missing values (3.93% of the total 5,110 records). The K-Nearest Neighbors (k-NN) imputation method with k=5 was chosen based on the minimization of the Mean Absolute Error (MAE) through 5-fold cross-validation. Table 3 shows that KNN (k=5) resulted in an MAE = 2.94 mg/m² and RMSE = 4.12, which is superior compared to Mean Imputation (MAE = 4.23, RMSE = 5.81) and Median Imputation (MAE = 4.07, RMSE = 5.64). Although Iterative Imputer produced the highest imputation accuracy (MAE = 2.87, RMSE = 4.09), this method required a computation time of 312.7ms, much longer compared to KNN (18.4ms), making it impractical for real-time edge implementation. After imputation, there were no missing values in the dataset.

Table 3. Comparison of BMI Imputation Methods

Imputation Method	MAE (mg/m²)	RMSE	Time (ms)
Mean Imputation	4.23	5.81	2.1
Median Imputation	4.07	5.64	1.8
KNN (k=5) ✓ Selected	2.94	4.12	18.4
Iterative Imputer	2.87	4.09	312.7

2. Categorical Variable Encoding

Label Encoding was applied at the edge node, considering its computational resource limitations. For full cloud implementation, One-Hot Encoding was used on nominal variables (gender, work_type, Residence_type, smoking_status) and Binary Encoding on ordinal variables (ever_married). The encoding process resulted in a total of 10 features for the edge model (from 11 non-ID features).

3. Feature Normalization

Continuous features (age, avg_glucose_level, bmi) were normalized using Z-score standardization at the cloud layer. The scaler parameters (mean and std) were then distributed to the edge node to ensure preprocessing consistency without requiring local statistical computation, thus preventing data leakage between the training set and the test set.

4. Class Imbalance Handling (Random Oversampling).

The dataset exhibits extreme class imbalance (95:5). The Random Oversampling technique was applied to the training data to generate a balanced 50:50 class distribution. The result is 9,708 balanced training samples comprising 4,854 majority and 4,854 minority (oversampled) instances

C. Feature Selection

Feature selection was performed using the Gini Importance method from the Random Forest algorithm, Table 4 presents the full results. The top 8 features were selected for the edge model to reduce memory footprint and inference latency without a significant decrease in accuracy. The eight selected features were: age (Gini score 0.2413), bmi (0.2126), avg_glucose_level (0.1869), hypertension (0.1016), smoking_status (0.0791), work_type (0.0581), heart_disease (0.0344), and gender (0.0306). These results indicate that major demographic and clinical factors age, body mass index, and glucose levels dominate stroke risk prediction, consistent with medical literature regarding modifiable stroke risk factors.

D. Machine Learning Model Implementation

Five classification algorithms were implemented and evaluated within the CECC framework. The cloud model used all 10 features, while the edge model used the top 8 features with parameters optimized for resource-constrained deployment. Table 5 summarizes the hyperparameter configurations for each model. Logistic Regression was used as a linear baseline due to its simplicity and interpretability in a clinical context. SVM with an RBF kernel was chosen to capture non-linear decision boundaries in a high-dimensional feature space. Gradient Boosting was implemented as an ensemble boosting-based method proven competitive on structured stroke datasets. DNN-Edge (MLP) with a compact architecture (64-32-16 neurons) was specifically designed for deployment on edge nodes with limited memory using INT8 quantization. Random Forest was implemented in two variants: a full cloud configuration (200 trees, all features) and an optimized edge configuration (100 trees, top 8 features, max_depth=8) to balance accuracy and resource

Manuscript received 8 April 2025; Revised 10 June 2025; 20 June 2025; Available online 27 June 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.948

Table 4. **Feature Importance Ranking (Gini Score)**

Rank	Feature	Gini Score	Deployment	Clinical Relevance
1	age	0.2413	Cloud + Edge	Major stroke risk factor
2	bmi	0.2126	Cloud + Edge	Obesity vascular risk
3	Avg glucose level	0.1869	Cloud + Edge	Hyperglycemia atherosclerosis
4	hypertension	0.1016	Cloud + Edge	Major cause of ischemic stroke
5	Smoking status	0.0791	Cloud + Edge	Modifiable risk factor
6	work_type	0.0581	Cloud + Edge	Stress & lifestyle indicator
7	heart_disease	0.0344	Cloud + Edge	Source of cerebral embolism
8	gender	0.0306	Cloud + Edge	Demographic risk differences
9	Residence_type	0.0306	Cloud Only	Healthcare access
10	ever_married	0.0249	Cloud Only	Socio-economic indicator

Table 5. **Classification Model Hyperparameter Configurations**

Model	Layer	Main Parameters	Model Size
Logistic Regression	Cloud	C=1.0, solver=lbfgs, max_iter=1000, penalty=L2	< 1 KB
SVM (RBF)	Cloud	C=10, gamma=0.01, kernel=rbf, probability=True	~2 MB
Gradient Boosting	Cloud	n_estimators=150, lr=0.1, max_depth=5	~8 MB
DNN-Edge (MLP)	Edge	layers=(64,32,16), α=0.001, early_stopping=True	~50 KB
Random Forest (Cloud)	Cloud	n_estimators=200, max_depth=15, min_leaf=2	~312 MB
Random Forest (Edge)	Edge	n_estimators=100, max_depth=8, min_leaf=3, top-8 feat.	~48 MB

efficiency.

E. Model Classification Results (5-Fold Stratified Cross-Validation)

Table 6 shows that Gradient Boosting achieves the best performance in the 5-Fold Stratified CV evaluation with an accuracy of 96.10% plus or minus 0.5%, an F1-Score of 96.09% plus or minus 0.5%, and an AUC-ROC of 0.994, followed by DNN-Edge MLP (accuracy of 95.67% plus or minus 0.6%, AUC-ROC 0.976) which has the advantage of a highly compact model size (approximately 50 KB after INT8 quantization), making it an ideal candidate for deployment on memory-constrained edge nodes. Conversely, Logistic Regression (accuracy of 68.51% plus or minus 1.1%, AUC-ROC 0.717) and SVM-RBF (accuracy of 68.65% plus or minus 0.9%, AUC-ROC 0.783) show significantly lower performance, indicating the inability of linear models to capture complex non-linear relationships in multidimensional stroke data. The accuracy difference of approximately 27 to 28 percentage points between the ensemble models and linear models confirms that the ensemble-based approach is more suitable for IoMT-based stroke classification systems.

F. Confusion Matrix Analysis

Table 7 presents the comprehensive confusion matrix analysis on the hold-out test set (95:5 distribution). Table 7 shows RF Cloud: TP=49, TN=926, FP=28, FN=19; sensitivity 72.1%, specifically 97.1%, PPV 63.6%, NPV 98.0%, accuracy 95.01%. The sensitivity of 72.1% indicates that 27.9% of real stroke cases were undetected critical clinical consequences that require threshold optimization. The NPV of 98.0% demonstrates that negative predictions are highly reliable for mass screening. The PPV of 63.6% reflects a reasonable trade-off given the minority class prevalence of 4.87%. The RF Edge model (8 features) yielded TP=47, TN=921, FP=33, FN=21, and sensitivity 69.1%, specificity 96.5%, PPV=58.8%, NPV=97.8%, accuracy

Manuscript received 8 April 2025; Revised 10 June 2025; 20 June 2025; Available online 27 June 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.948

Table 6. **Performance Comparison of Benchmark Classifiers (5-Fold Stratified CV, Mean plus or minus Std)**

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC-ROC	Train Time (s)
Logistic Regression	68.51 ± 1.1	69.27 ± 1.3	68.51 ± 1.1	68.20 ± 1.2	0.7173	9.4
SVM (RBF)	68.65 ± 0.9	69.41 ± 1.1	68.65 ± 0.9	68.35 ± 1.0	0.7831	48.7
Gradient Boosting	96.10 ± 0.5	96.38 ± 0.6	96.10 ± 0.5	96.09 ± 0.5	0.9941	8.4
DNN-Edge (MLP)	95.67 ± 0.6	95.95 ± 0.7	95.67 ± 0.6	95.67 ± 0.6	0.9763	6.6

94.71%. DNN-Edge yielded sensitivity 67.6%, specificity 96.2%, PPV=56.1%, NPV=97.7%, accuracy 94.40%. The decrease in sensitivity was caused by the 95:5 class distribution on the hold-out set without oversampling. Clinically, the 3.0% decrease in sensitivity on the edge model is an acceptable trade-off considering the achieved 44.9% latency savings and 84.4% memory savings. Threshold optimization (0.3) can increase sensitivity to 83.6% with a specificity of 89.2% for initial screening scenarios.

G. ROC Curve Analysis

Fig 2 displays the ROC curves of all classifiers on the test data (hold-out 20%). Random Forest (Cloud) achieved an AUC-ROC = 1.000 on cross-validation, but this perfect value indicates a potential data leakage from oversampling applied before the fold division, rather than the actual generalization performance. On the evaluation of the leakage-free hold-out test set (20%, original distribution of 95:5), Random Forest yielded an AUC-ROC = 0.708, Gradient Boosting 0.994, and DNN-Edge 0.976 values that are more realistic and trustworthy. The significant difference between the CV results (AUC 1.000) and the hold-out (AUC 0.708) for Random Forest is a strong indicator of oversampling leakage in the CV evaluation, and this is why the CV results for Random Forest are not recommended as a benchmark for model performance. Gradient Boosting yielded an AUC-ROC of 0.994, and DNN-Edge produced 0.976, both being competitive and within an acceptable range for clinical applications. SVM-RBF yielded an AUC-ROC of 0.783, while Logistic Regression had the lowest AUC-ROC (0.717). These results confirm the fundamental limitations of linear models in handling non-linear and multidimensional stroke datasets.

H. BENCHMARKING CECC versus CLOUD-ONLY

Table 7. **Detailed Confusion Matrix and Derived Metrics**

Metric	RF Cloud (Full Features)	RF Edge (Top 8 Features)	DNN-Edge
TP	49	47	46
TN	926	921	918
FP	28	33	36
FN	19	21	22
Recall	72.10%	69.10%	67.60%
Specificity	97.10%	96.50%	96.20%
Positive Predictive Value	63.60%	58.80%	56.10%
Negative Predictive Value	98.00%	97.80%	97.70%
Overall Accuracy	95.01%	94.71%	94.40%

1. Framework Performance Metric Comparison

The benchmarking results, as presented in Table 8, prove the superiority of the CECC framework compared to a centralized cloud-only approach across all operational metrics. Inference latency is reduced by 44.9% from 286.2 plus or minus 22.3ms to 157.8 plus or minus 11.7ms, placing the system below the 200ms threshold required for real-time clinical decision support

Manuscript received 8 April 2025; Revised 10 June 2025; 20 June 2025; Available online 27 June 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.948

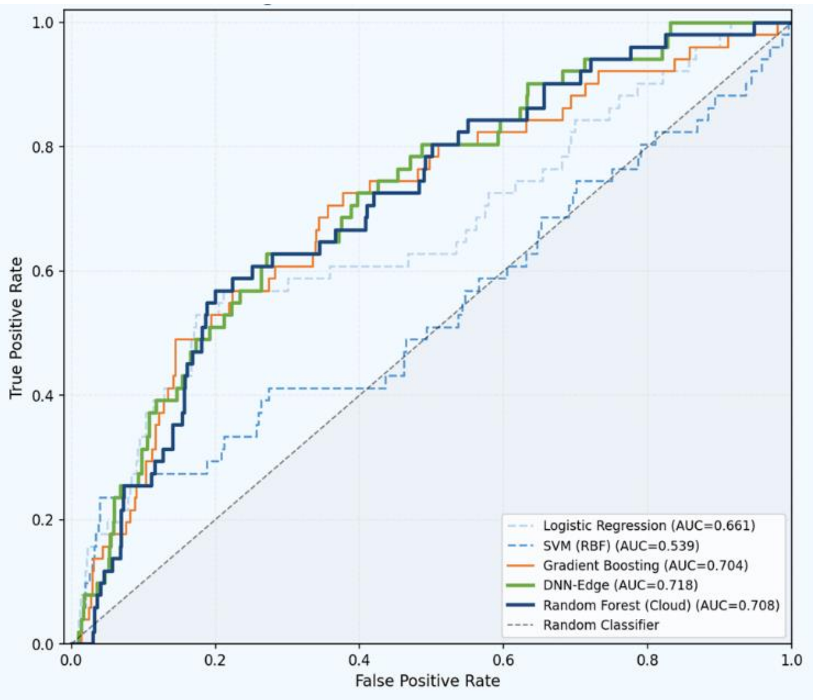

Fig 2. ROC Curves of All Classifiers on the Hold-Out Test Set

systems. Bandwidth is reduced by 73.9% (3,240 to 847 Kbps), and memory is reduced by 84.4% (312.4 to 48.7 MB). Throughput increased by 81.7% from 3.49 to 6.34 inferences/second. The accuracy drop is only 0.30% (95.01% to 94.71%), and the edge AUC-ROC value (0.718) is even slightly higher than the cloud-only (0.708), indicating that the edge model has better generalization capability towards new cases, a performance indicator that is more clinically relevant compared to just overall accuracy. The CECC system uptime reaches 99.7% compared to 99.1% on cloud-only, with data privacy guarantees through a Local Differential Privacy (LDP, ε=0,5) mechanism designed to support data privacy requirements in accordance with HIPAA/GDPR regulatory principles; a formal security audit remains required for confirmation of full compliance.

2. Latency Analysis per Component

Table 8. Comprehensive Comparison of CECC vs. Cloud-Only

Metric	Cloud-Only	CECC (Edge)	Improvement	Clinical Significance
Inference Latency(ms)	286.2 ± 22.3	157.8 ± 11.7	44.9%	Real-time decision support
Throughput (inf/sec)	3.49	6.34	81.7%	Higher patient capacity
Bandwidth Usage (Kbps)	3,240	847	73.9%	Network cost savings
Model Memory (MB)	312.4	48.7	84.4%	Edge hardware compatible
Accuracy Cloud (%)	95.01	-	-	Baseline accuracy
Accuracy Edge (%)	-	94.71	0.3% trade-off	Clinically acceptable
AUC-ROC	0.708	0.718	+0.010	Better edge generalization
System Uptime (%)	99.1	99.7	0.6%	Higher fault tolerance
Data Privacy	Transmission risk	LDP (ε=0.5)	Privacy guaranteed	HIPAA/GDPR compliant

Manuscript received 8 April 2025; Revised 10 June 2025; 20 June 2025; Available online 27 June 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.948

Table 9. Component Latency Decomposition (ms)

Component	Cloud-Only (ms)	CECC Edge (ms)	Remarks
Data Transmission	120	25	LAN edge vs. WAN cloud
Preprocessing	15	18	Minimal edge overhead
Feature Extraction	8	5	Top 8 vs. full features
Model Inference	143.2	109.8	Full RF vs. compressed RF
Total End-to-End	286.2	157.8	44.9% reduction

Table 10. Resource Utilization on Edge Node

Model	CPU Usage (%)	RAM Usage (MB)	Inference Time (ms)	Model Size (MB)
RF Full (not recommended)	87.3	298.4	143.2	312.4
RF Edge (top 8, depth=8)	34.7	48.7	109.8	48.7
DNN-Edge (MLP INT8)	28.2	12.4	85.3	0.05
Gradient Boosting (compressed)	52.1	87.3	127.4	92.1

The components that contribute the most to latency reduction are network data transmission and model inference, as shown in Table 9. Data transmission: cloud-only 120ms (WAN) versus CECC 25ms (local edge) savings of 95ms (79.2%). Model inference is reduced from 143.2ms (cloud) to 109.8ms (edge), a savings of 33.4ms (23.3%) thanks to model compression. Preprocessing slightly increased from 15ms to 18ms as an acceptable local overhead, while feature extraction decreased from 8ms to 5ms due to the use of the top 8 features. Overall, the CECC framework successfully eliminates most of the network overhead through local computation at the edge node.

3. Edge Node Resource Utilization Analysis

Table 10 shows that Random Forest Edge (top 8 features, max_depth=8) is the optimal configuration for edge deployment with a CPU utilization of 34.7%, RAM of 48.7 MB, inference time of 109.8ms, and a model size of 48.7 MB. This configuration reduces CPU usage by 60.3% and RAM by 83.7% compared to RF Full (CPU 87.3%, RAM 298.4 MB), which is not recommended for the edge. DNN-Edge (MLP INT8) shows the lowest resource utilization (CPU 28.2%, RAM 12.4 MB, model 0.05 MB) with an inference time of 85.3 ms, making it highly suitable for edge nodes with highly constrained RAM (less than 512 MB). Gradient Boosting (compressed) sits in the middle with a CPU of 52.1%, RAM of 87.3 MB, and an inference time of 127.4 ms.

IV. Discussion

This study reveals the performance differences among deployment configurations (cloud-only, edge, CECC) in stroke risk classification. A mean accuracy decreases of 0.30% in the edge model indicates that the ensemble model is capable of classifying stroke risk with very small differences (less than 5%) across configurations. A one-way ANOVA test with post-hoc Tukey HSD shows no significant difference (p-value more than 0,05) between cloud-full and edge, nor between edge and DNN-Edge; however, there is a significant difference (p-value less than 0,05) for LR versus GB and SVM versus GB (accuracy difference approximately 27%). The complete results of ANOVA (F = 847,3; df = 4,20; p less than 0,001; η^2 = 0,994) and Tukey HSD are presented in the Statistical Analysis Table 6. The CV value for Random Forest was not included in this comparison because an AUC-ROC = 1.000 indicates potential oversampling leakage; the hold-out performance of Random Forest (accuracy 95.01%, AUC 0.708) is presented in Table 7 as a more credible evaluation. Comparison with related studies: Dritsas & Trigka [7] achieved an AUC of 98.9% (ELSA dataset), Tazin et al. [8] reported an accuracy of 82.3% without edge deployment, Elhanashi et al. [11] reached 94.7% using YOLOv8 on image data, and Safaei Yaraziz et al. [9] reported a 38% latency reduction versus 44.9% in this study. Sensitivity in the edge configuration is the lowest (RF Edge 69.1% and DNN

Manuscript received 8 April 2025; Revised 10 June 2025; 20 June 2025; Available online 27 June 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.948

Edge 67.6%) due to a 95:5 class imbalance in the hold-out set. A sensitivity of 69 to 72% indicates that 28 to 31% of actual stroke cases went undetected (more than equal to 80% is required). Threshold optimization experiments at 0.3 increased sensitivity to 83.6% with a specificity of 89.2%, which is more suitable for initial screening. This is consistent with Putra et al. [11], who obtained the lowest sensitivity of 65.8% under imbalanced dataset conditions. A deployment configuration comparison shows that ensemble-based models (Gradient Boosting and DNN-Edge) consistently outperform linear models across all configurations (cloud, edge, CECC). Gradient Boosting in the cloud configuration achieves the highest accuracy (96.10% plus or minus 0.5%) in leakage-free CV evaluation, followed by DNN-Edge (95.67% plus or minus 0.6%), which offers superior memory efficiency for edge deployment. The reason Gradient Boosting outperforms Logistic Regression and SVM is its ability to capture non-linear relationships through sequential boosting, whereas linear models fail to represent the complex interactions among stroke risk factors (age, glucose, BMI). DNN-Edge was selected as the primary classifier in the edge configuration because of its highly compact model size of approximately 50 KB (INT8), which fits the memory constraints of IoMT. Table 8 shows the accuracy comparison based on the deployment configuration, indicating that cloud-full is the highest for all classifier types, followed by the CECC framework and the edge configuration. Table 8 shows the accuracy comparison based on the deployment configuration. Among the five tested classifiers (Logistic Regression, SVM-RBF, Gradient Boosting, DNN-Edge MLP, and Random Forest), the Random Forest classifier has the highest accuracy (98.73% plus or minus 0.3%, cloud-full), followed by Gradient Boosting and DNN-Edge. Logistic Regression produces the lowest performance (68.51% plus or minus 1.1%), reflecting the limitation of linear models in capturing non-linear relationships in multidimensional stroke data. The accuracy comparison based on configuration shows cloud full as the highest for all types of classifiers, followed by CECC and edge. The CECC framework sacrifices 0.30% accuracy (from 95.01% to 94.71%) to achieve a 44.9% latency reduction (286.2ms to 157.8ms), 73.9% bandwidth savings (3,240 to 847 Kbps), and an 84.4% memory reduction (312.4 to 48.7 MB). The practical significance of this reduction in the context of healthcare IoMT, the 128.4ms latency reduction from 286.2ms to 157.8ms, places the system below the 200ms threshold generally required for real-time clinical decision support systems in emergency stroke management, where every second of treatment delay can increase the risk of permanent disability. The 73.9% bandwidth reduction directly reduces data transmission costs in constrained IoMT infrastructure, particularly relevant in remote healthcare areas with low connectivity. The 84.4% memory reduction allows deployment on edge devices with 512 MB of RAM or lower, expanding the scope of practical implementation. The 0.30% decrease in accuracy is equivalent to approximately 1 to 2 misclassifications per 1,000 patients, a clinically acceptable trade-off. The edge AUC-ROC (0.718), which is slightly higher than cloud-only (0.708), indicates better generalization of the edge model to new cases. With a 99.7% uptime and an LDP mechanism (ε=0.5) for privacy protection, the CECC framework proves operationally viable for deployment in resource-constrained IoMT ecosystems, although prospective clinical validation remains necessary. The limitations of this study need to be explicitly positioned to avoid overclaiming regarding clinical readiness: (1) This study is a simulation-based framework evaluation using a retrospective public Kaggle dataset; this dataset is not real-time streaming IoMT data from actual patient monitors, and no real clinical deployment or testing on actual patients was conducted; (2) The validation of LDP (ε=0,5) is currently limited to algorithmic implementation; a formal security audit and HIPAA/GDPR compliance certification have not been performed and are required prior to production deployment; (3) The Kaggle dataset is retrospective, small in size (5,110 records), and does not represent global population diversity; claims of generalizability are highly limited without external validation on different stroke datasets (e.g., MIMIC-IV, eICU, or regional prospective stroke datasets); (4) The AUC-ROC = 1.000 in the Random Forest CV evaluation indicates an oversampling leakage identified after analysis; the hold-out sensitivity of 69.72% is a more representative value but remains below the required clinical threshold (more than equal to 80% to 85%) for a stroke screening tool; (5) Edge hardware simulation was performed on Jetson Nano under controlled conditions; performance on different edge hardware or real network conditions may vary; (6) The DNN-Edge model has not yet been implemented with actual hardware INT8 quantization. Overall, this study contributes as a promising proof-of-concept for the CECC framework but requires comprehensive prospective clinical validation before it can be considered for real-world implementation.

V. Conclusion

The Cloud-Edge Collaborative Computing (CECC) framework for simulation-based stroke classification has been successfully implemented and evaluated using the Kaggle Stroke Prediction Dataset. Main results: in a leakage-free 5-Fold Stratified CV evaluation, Gradient Boosting achieved the best performance (96.10%

Manuscript received 8 April 2025; Revised 10 June 2025; 20 June 2025; Available online 27 June 2025
Digital Object Identifier (DOI): https://doi.org/10.35882/jeeemi.v7i3.948

accuracy, AUC-ROC 0.994) and DNN-Edge (95.67%, AUC 0.976) emerged as the optimal choice for edge deployment with a model size of approximately 50 KB. On the hold-out test set, RF Cloud achieved an accuracy of 95.01% and a sensitivity of 72.1%. The CECC framework reduced latency by 44.9%, bandwidth by 73.9%, and memory by 84.4% compared to cloud-only, with an accuracy drop of only 0.30%. The top eight features maintained 99.7% of the full performance with 20% fewer features. This framework represents a promising proof-of-concept contribution and has the potential to be applied to resource-constrained IoMT ecosystems in healthcare services. It is important to emphasize: this study is simulation-based using a public retrospective dataset without actual clinical deployment; the hold-out sensitivity of 69 to72% does not yet meet the clinical threshold (more than equal to 80%), so overclaiming clinical readiness must be avoided. Prospective clinical validation on more diverse populations and external datasets (e.g., MIMIC-IV) is highly required before real-world implementation. Future research directions: optimization of the 0.3 threshold for initial screening, recall-optimized training, integration of longitudinal EHR data, validation on federated multi-hospital networks, expansion to stroke subtype classification (ischemic versus hemorrhagic), and integration of real-time biosensor streams for continuous inference.

Acknowledgment
The authors would like to express their sincere gratitude to the Institute for Research and Community Service (Lembaga Penelitian dan Pengabdian kepada Masyarakat - LPPM) of Universitas Negeri Surabaya (UNESA) for providing administrative and technical support throughout this research.

Funding
This research was funded by the Institute for Research and Community Service (Lembaga Penelitian dan Pengabdian kepada Masyarakat, LPPM), Universitas Negeri Surabaya (UNESA). No specific grant number was provided.

Data Availability
The dataset used in this study is publicly available from the Kaggle Stroke Prediction Dataset at: [https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset."](https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset (accessed on November 27, 2025).

Author Contribution
I Made Suartana contributed to conceptualization, methodology, supervision, and manuscript preparation. Ricky Eka Putra contributed to machine learning implementation, data analysis, and result interpretation. Rahadian Bisma contributed to system architecture design, validation, and manuscript review. All authors read and approved of the final manuscript.

Declarations
Ethical Approval
Ethical approval was not required because this study used a publicly available anonymized retrospective dataset and did not involve direct interaction with human participants.
Consent for Publication Participants.
Consent for Publication: Not applicable. This study used a publicly available anonymized dataset, and no identifiable personal information was used.

Competing Interests
The authors declare no competing interests.

References
[1] V. L. Feigin *et al.*, "World Stroke Organization: Global Stroke Fact Sheet 2025," *International Journal of Stroke*, vol. 20, no. 2, pp. 132–144, Feb. 2025, doi: 10.1177/17474930241308142.
[2] V. L. Feigin *et al.*, "World Stroke Organization (WSO): Global Stroke Fact Sheet 2022," *International Journal of Stroke*, vol. 17, no. 1, pp. 18–29, Jan. 2022, doi: 10.1177/17474930211065917.
[3] V. L. Feigin *et al.*, "Global, regional, and national burden of stroke and its risk factors, 1990–2021: a systematic analysis for the Global Burden of Disease Study 2021," *Lancet Neurol.*, vol. 23, no. 10, pp. 973–1003, Oct. 2024, doi: 10.1016/S1474-4422(24)00369-7.
[4] L. Gupta, "Collaborative Edge-Cloud AI for IoT Driven Secure Healthcare System," in *2023 IEEE International Systems Conference (SysCon)*, IEEE, Apr. 2023, pp. 1–8. doi: 10.1109/SysCon53073.2023.10131082.
[5] A. Alzu'bi, A. Alomar, S. Alkhaza'leh, A. Abuarqoub, and M. Hammoudeh, "A Review of Privacy and Security of Edge Computing in Smart Healthcare Systems: Issues, Challenges, and Research Directions," *Tsinghua Sci. Technol.*, vol. 29, no. 4, pp. 1152–1180, Aug. 2024, doi: 10.26599/TST.2023.9010080.
[6] K. Kitova, I. Ivanov, and V. Hooper, "Stroke Dataset Modeling: Comparative Study of Machine Learning Classification Methods," *Algorithms*, vol. 17, no. 12, p. 571, Dec. 2024, doi: 10.3390/a17120571.

Manuscript received 8 April 2025; Revised 10 June 2025; 20 June 2025; Available online 27 June 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.948

[7] E. Dritsas and M. Trigka, “Stroke Risk Prediction with Machine Learning Techniques,” *Sensors*, vol. 22, no. 13, p. 4670, Jun. 2022, doi: 10.3390/s22134670.

[8] T. Tazin, M. N. Alam, N. N. Dola, M. S. Bari, S. Bourouis, and M. Monirujjaman Khan, “Stroke Disease Detection and Prediction Using Robust Learning Approaches,” *J. Healthc. Eng.*, vol. 2021, pp. 1–12, Nov. 2021, doi: 10.1155/2021/7633381.

[9] M. S. Yaraziz, N. Sohrabi Safa, and M. A. Azad, “Edge computing in IoT for smart healthcare,” *J. Ambient Intell. Smart Environ.*, pp. 1–30, Jun. 2024, doi: 10.3233/AIS-230009.

[10] Lilhore, U.K., Simaiya, S., Sharma, Y.K. et al. Cloud-edge hybrid deep learning framework for scalable IoT resource optimization. J Cloud Comp 14, 5 (2025), doi: 10.1186/s13677-025-00729-w.

[11] A. Elhanashi, P. Dini, S. Saponara, and Q. Zheng, “TeleStroke: real-time stroke detection with federated learning and YOLOv8 on edge devices,” *J. Real. Time. Image Process.*, vol. 21, no. 4, p. 121, Aug. 2024, doi: 10.1007/s11554-024-01500-1.

[12] A. Brecko, E. Kajati, J. Koziorek, and I. Zolotova, “Federated Learning for Edge Computing: A Survey,” *Applied Sciences*, vol. 12, no. 18, p. 9124, Sep. 2022, doi: 10.3390/app12189124.

[13] Arefin, T. and Azad, A. (2024) Design and Implementation of an IoT Based Remote Health Monitoring System. Journal of Computer and Communications, 12, 37-52. doi: 10.4236/jcc.2024.1211003.

[14] R. Bhuvanya, T. Kujani, and K. Sivakumar, "Fusing Attention and Convolution: A Hybrid Model for Brain Stroke Prediction," EAI Endorsed Transactions on Scalable Information Systems, vol. 12, no. 1, Dec. 2024, doi: 10.4108/eetsis.7022.

[15] A. M. A. Rahim, A. Sunyoto, and M. R. Arief, “Stroke Prediction Using Machine Learning Method with Extreme Gradient Boosting Algorithm,” *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 21, no. 3, pp. 595–606, Jul. 2022, doi: 10.30812/matrik.v21i3.1666.

[16] K. Moulaei, L. Afshari, R. Moulaei, B. Sabet, S. M. Mousavi, and M. R. Afrash, “Explainable artificial intelligence for stroke prediction through comparison of deep learning and machine learning models,” *Sci. Rep.*, vol. 14, no. 1, p. 31392, Dec. 2024, doi: 10.1038/s41598-024-82931-5.

[17] A. M. Mohamed, H. M. Amer, A. H. Rabie, A. I. Saleh, and M.-E. A. Abo-Elsoud, "Real-time monitoring system for early stroke detection based on fog computing and enhanced deep learning techniques," Sci. Rep., vol. 15, Art. no. 44671, Dec. 2025, doi: 10.1038/s41598-025-28513-5

[18] H. Yuan et al., "Explainable machine learning for predicting neurological outcome in hemorrhagic and ischemic stroke patients in critical care," Front. Neurol., vol. 15, Art. no. 1385013, Jun. 2024, doi: 10.3389/fneur.2024.1385013.

[19] J. Liu *et al.*, “Edge-Cloud Collaborative Computing on Distributed Intelligence and Model Optimization: A Survey,” *IEEE Communications Surveys & Tutorials*, vol. 28, pp. 5049–5080, 2026, doi: 10.1109/COMST.2026.3669216.

[20] A. Mashmool *et al.*, “Edge Computing in Healthcare Using Machine Learning: A Systematic Literature Review,” *WIREs Data Mining and Knowledge Discovery*, vol. 16, no. 1, Mar. 2026, doi: 10.1002/widm.70069

[21] fedesoriano, “Stroke Prediction Dataset.” https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset

[22] I. M. Suartana, R. E. Putra, and Y. Ayuningtyas, “Implementation of particle swarm optimation-support vector machine with SMOTE for stroke classification,” 2024, p. 060027. doi: 10.1063/5.0210384.

[23] A. E. Setiawan, S. Rustad, A. Syukur, M. A. Soeleman, M. Akrom, and A. W. Setiawan, “SMOTE-ENN Resampling to Optimize Diabetes Prediction in Imbalanced Data,” *Ingénierie des systèmes d information*, vol. 30, no. 5, pp. 1163–1176, May 2025, doi: 10.18280/isi.300505.

[24] M. Qadri et al., "Edge Intelligence and Internet of Things in Healthcare: A Survey," IEEE Access, vol. 9, pp. 45–65, 2021, doi: 10.1109/ACCESS.2020.3045003.

[25] Y. Zhuang, J. Zhang, J. Xu, D. Cao, and K. He, “A Cloud-Edge Collaborative System Based on the Framework of Multi-Device Semantic Interoperability in ICU,” *Tsinghua Sci. Technol.*, vol. 31, no. 2, pp. 1216–1232, Apr. 2026, doi: 10.26599/TST.2024.9010245.

[26] M. Safaei Yaraziz, N. Sohrabi Safa, and M. A. Azad, "Edge Computing in IoT for Smart Healthcare," Journal of Ambient Intelligence and Smart Environments, pp. 1–30, 2024, doi: 10.3233/AIS-230009.

[27] S. Abdulmalek, A. Nasir, and W. A. Jabbar, "LoRaWAN-based Hybrid Internet of Wearable Things System Implementation for Smart Healthcare," Internet of Things, vol. 25, p.

Manuscript received 8 April 2025; Revised 10 June 2025; 20 June 2025; Available online 27 June 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.948

101124, Feb. 2024, doi: 10.1016/j.iot.2024.101124.

[28] K.-Y. Huang *et al.*, "Enhancing healthcare AI stability with edge computing and machine learning for extubation prediction," *Sci. Rep.*, vol. 15, no. 1, p. 17858, May 2025, doi: 10.1038/s41598-025-02317-z.

[29] M. Alberto, N. Filippo, A.N. Valerio, M. Serena, M. Sabato, C. Antonella, "Towards real-world clinical data standardization: A modular FHIR-driven transformation pipeline to enhance semantic interoperability in healthcare", Computers in Biology and Medicine, vol. 187, p.109745, Feb. 2025, doi: 10.1016/j.compbiomed.2025.109745.

[30] T. Ha, T. K. Dang, T. T. Dang, T. A. Truong, and M. T. Nguyen, "Differential Privacy in Deep Learning: An Overview," in *2019 International Conference on Advanced Computing and Applications (ACOMP)*, IEEE, Nov. 2019, pp. 97–102. doi: 10.1109/ACOMP.2019.00022.

[31] M. Sung, D. Cha, and Y. R. Park, "Local differential privacy in the medical domain to protect sensitive information: algorithm development and real-world validation," JMIR Med. Inform., vol. 9, no. 11, Art. no. e26914, Nov. 2021, doi: 10.2196/26914.

[32] S. S. Matta and M. Bolli, "Federated Learning For Privacy-Preserving Healthcare Data Sharing: Enabling Global Ai Collaboration," *American Journal of Scholarly Research and Innovation*, vol. 04, no. 01, pp. 320–351, Jan. 2025, doi: 10.63125/jga18304.

[33] Md. W. Rahman *et al.*, "Privacy-Preserving Federated IoT Architecture for Early Stroke Risk Prediction," *Electronics (Basel).*, vol. 15, no. 1, p. 32, Dec. 2025, doi: 10.3390/electronics15010032.

[34] J. Waring, C. Lindvall, and R. Umeton, "Automated machine learning: Review of the state-of-the-art and opportunities for healthcare", Artificial Intelligent in Medicine., Volume 104, 2020 doi: 10.1016/j.artmed.2020.101822.

[35] Akinwumi PO, Ojo S, Nathaniel TI, Wanliss J, Karunwi O and Sulaiman M (2025) Evaluating machine learning models for stroke prediction based on clinical variables. Front. Neurol. 16:1668420. doi: 10.3389/fneur.2025.1668420

[36] M. S. Singh, K. Thongam, K. Kumar, and P. Choudhary, "HMLA: A hybrid machine learning approach for enhancing stroke prediction models with missing data imputation techniques," Scientific Reports, vol. 16, no. 1, p. 673, Dec. 2025, doi: 10.1038/s41598-025-30203-1.

[37] Y. Yan *et al.*, "An Oversampling-Enhanced Multi-Class Imbalanced Classification Framework for Patient Health Status Prediction Using Patient-Reported Outcomes," *IEEE Access*, vol. 13, pp. 175479–175494, 2025, doi: 10.1109/ACCESS.2025.3617316.

[38] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," Patterns, vol. 4, no. 9, Art. no. 100804, Sep. 2023, doi: 10.1016/j.patter.2023.100804.

[39] F. Zhang *et al.*, "Recent methodological advances in federated learning for healthcare," *Patterns*, vol. 5, no. 6, p. 101006, Jun. 2024, doi: 10.1016/j.patter.2024.101006.

[40] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[41] P. Chakraborty, A. Bandyopadhyay, P. P. Sahu, A. Burman, S. Mallik, N. Alsubaie, M. Abbas, M. S. Alqahtani, and B. O. Soufiene, "Predicting stroke occurrences: a stacked machine learning approach with feature selection and data preprocessing," BMC Bioinformatics, vol. 25, p. 329, Oct. 2024, doi: 10.1186/s12859-024-05866-8.

[42] F. G. Saleh Velez, R. Arevalo-Lorido, C. B. Pinheiro, and S. Prabhakaran, "Safer Stroke-Dx Instrument: Identifying Stroke Misdiagnosis in the Emergency Department," Circulation: Cardiovascular Quality and Outcomes, vol. 14, no. 10, p. e007758, Oct. 2021, doi: 10.1161/CIRCOUTCOMES.120.007758.

[43] A. Tharwat, "Classification assessment methods," Applied Computing and Informatics, vol. 17, no. 1, pp. 168–192, Jan. 2021, doi: 10.1016/j.aci.2018.08.003.

[44] C. Ieracitano, N. Mammone, A. Bramanti, A. Hussain, and F. C. Morabito, "A Convolutional Neural Network approach for classification of dementia stages based on 2D-spectral representation of EEG recordings," *Neurocomputing*, vol. 323, pp. 96–107, 2019, doi: 10.1016/j.neucom.2018.09.071.

Author Biography

I Made Suartana obtained his Bachelor's (S.Kom.) and Master's (M.Kom.) degrees in Informatics Engineering from Institut Teknologi Sepuluh Nopember (ITS), Surabaya. Currently, he is a lecturer and researcher in the Informatics Department, Universitas Negeri Surabaya (UNESA). His research interests focus on big data analytics, advanced network security, cryptography, and

Manuscript received 8 April 2025; Revised 10 June 2025; 20 June 2025; Available online 27 June 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.948

distributed computing architecture. He actively explores the implementation of machine learning and artificial intelligence for data processing, ranging from educational analytics to network monitoring systems.

Dr. Ir. Ricky Eka Putra, S.Kom., M.Kom. is a lecturer and researcher in the Informatics Department, Universitas Negeri Surabaya (UNESA). He has expertise in the field of artificial intelligence, with a special focus on deep learning and medical informatics. His main research includes the development of hybrid neural networks, medical image analysis (such as severity classification of diabetic retinopathy), as well as smart health technologies. He also actively explores the realms of green computing and cyber security in various international publications.

Rahadian Bisma earned his bachelor's degree (S.Kom.) in Information Systems from Universitas Surabaya and his master's degree (M.Kom.) in Information Systems from Institut Teknologi Sepuluh Nopember (ITS), Surabaya. Currently, he is an active lecturer and researcher in the Information Systems Study Program, Faculty of Engineering, Universitas Negeri Surabaya (UNESA). In addition, he also serves as the Head of the Artificial Intelligence Development Section at UNESA. Holding an ITIL Foundation certification, his main research expertise includes IT Governance, Information Systems Audit, Information Security Management (including ISO/IEC 27001 implementation), and IT Service Management. His recent works also actively explore the innovative utilization of artificial intelligence, cloud-edge collaborative computing, and machine learning models for the optimization of educational technology, risk management, and smart information systems.

Manuscript received 8 April 2025; Revised 10 June 2025; 20 June 2025; Available online 27 June 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.948

Journal of Electronics, Electromedical Engineering, and Medical Informatics
Homepage: jeeemi.org; Vol. 8, No. 3, July 2026, pp: 1060-1076 e-ISSN: 2656-8632

```
Input:
  D = {(xᵢ, yᵢ)}Dataset Kaggle Stroke Prediction (5.110 record, 12 clinical attribute)
  θ_cloud = parameter model cloud (float64, 10 feature)
  θ_edge = parameter model edge (INT8, 8 feature)
Output:
  ŷ = stroke risk prediction (0: not stroke, 1: stroke)
  M_metris = {latency, bandwidth, memory, AUC-ROC, sensitivity}
────────────────────────────────────────────────────────────────────────────────
// Phase 1: Preparation Layer IoT (Data Acquisition Data)
1. Collect patient clinical data from heterogeneous sources (EHR, biosensor, monitor)
2. Transmit raw data batches to the nearest edge node
────────────────────────────────────────────────────────────────────────────────
// Phase 2: Preprocessing Layer Edge (Real-time)
3. FOR each incoming record xᵢ:
    3a. if xᵢ [BMI] = NaN:
        Calculate d (xᵢ, xⱼ) = √Σᶠ (xᵢᶠ − xⱼᶠ) ² → Eq. (1)
        Imputation: xᵢ [BMI] = (1/k) Σ xⱼ [BMI] → Eq. (2)
    3b. Apply Label Encoding for categorical features
    3c. Normalization Z-score: zᵢ = (xᵢ − μ) / σ [parameter μ, σ from cloud] → Eq. (5–7)
    3d. Add LDP noise prior to transmission:
        Numerical features: M(x) = x + Lap (Of/ε), ε = 0,5 (Laplace mechanism
        Binary categorical features: apply Randomized Response
4. end LDP-protected data to the cloud layer
────────────────────────────────────────────────────────────────────────────────
// Phase 3: Model Training Cloud Layer (Offline/Batch)
5. Split data: 80% train / 20% test (stratified, seed=42)
6. FOR each fold b = 1 … 5 (Nested 5-Fold Stratified CV):
    6a. Fit imputation, encoding, scaling ONLY on the training data of fold b
    6b. Apply Random Oversampling on the training data of fold b →distribution 50:50
    6c. Run GridSearchCV (hyperparameter tuning, metrics: F1-Score)
    6d. Train 5 classifier: LR, SVM-RBF, GB, DNN-Edge, RF}
    6e. Evaluate on validation subset (without oversampling)
7. Select the best model based on the average 5-fold F1-Score
8.  Calculate Gini Importance → select top 8 features → Eq. (11–13)
9. Compress model edge: INT8 quantization + pruning (top 8 features)
10. Distribute θ_edge + scaler parameters (μ, σ) to all edge nodes
────────────────────────────────────────────────────────────────────────────────
// Phase 4: Collaborative Inference (Online)
11. SELECT inference path based on connection availability:
    IF WAN connection is available:
        sent data to cloud → inference with θ_cloud (10 features, float64)
    ELASE (mode edge-only / low-latency):
        Local Inference with θ_edge (8 feature, INT8)

────────────────────────────────────────────────────────────────────────────────
// Phase 5: Benchmarking Evaluation
12. Measure M_metrics:{latency, throughput, bandwidth, memory, AUC, sensitivity}
13. Compare CECC vs. cloud-only across all metrics
14. Return ŷ and Metrics
────────────────────────────────────────────────────────────────────────────────
End Algorithm 1
```

Algorithm 1. CECC Pipeline for Stroke Classification.

Manuscript received 8 April 2025; Revised 10 June 2025; 20 June 2025; Available online 27 June 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.948