International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 13– No.5, November 2026 – www.ijais.org

# A Recurrent Neural Network-based Architecture for Early Stroke Detection and Diagnosis using Electromyography (EMG) Data

Bob Chile-Agada
Department of Computer Science
University of Port Harcourt
Port Harcourt, Nigeria

Laud Charles Ochei
Department of Computer Science
University of Port Harcourt
Port Harcourt, Nigeria

Fubara Egbono
Department of Computer Science
University of Port Harcourt
Port Harcourt, Nigeria

## ABSTRACT
Stroke remains a leading cause of mortality and long-term disability worldwide and imposes a heavy economic burden on health systems, families and societies. Early detection is critical for timely intervention, yet current diagnostic pathways often depend on expensive imaging, suffer from high false-positive rates and scale poorly in resource-limited settings. This paper presents an integrated Recurrent Neural Network (RNN)-based architectural framework for the early detection and diagnosis of stroke from surface Electromyography (sEMG) data, together with a scenario-based evaluation of that framework. The architecture exploits the sequential learning capability of Gated Recurrent Unit (GRU) and CNN-GRU encoders to model the temporal patterns of multichannel EMG signals, and it combines a robust preprocessing and windowing pipeline with gradient clipping, learning-rate scheduling and class-weighted training to stabilise learning on small, imbalanced cohorts. The framework is evaluated with the Architecture Trade-off Analysis Method (ATAM) across eight quality-attribute scenarios, covering routine clinical screening, class imbalance, high-volume population screening, corrupted signals, offline rural deployment, Electronic Health Record (EHR) integration, model updating and patients with severe motor impairment. The evaluation identifies the sensitivity points, trade-offs, risks and non-risks of the design and the anticipated impact of architectural changes on accuracy, efficiency, scalability, robustness and interpretability. The contribution is an architectural design and its evaluation, not a clinical validation study; a subject-wise validation protocol for future empirical work is specified.

## General Terms
Artificial Intelligence, Health Informatics, Software Architecture, Biomedical Signal Processing.

## Keywords
Stroke detection, Electromyography (EMG), Recurrent Neural Networks (RNN), Gated Recurrent Unit, Early diagnosis, Architecture Trade-off Analysis Method (ATAM), Scenario-based evaluation, Healthcare informatics.

## 1. INTRODUCTION
Stroke is one of the most significant causes of mortality in the world and a critical public-health challenge, leaving survivors at risk of high medical costs and long-term disability [1]. Approximately 13.7 million new strokes occur worldwide each year and about 5.5 million people die as a result [2]. When direct healthcare costs and productivity losses are combined, the global economic burden of stroke is estimated to exceed US$1 trillion annually [3]. The ability to predict stroke risk is therefore valuable for screening and prevention, particularly in underserved regions where access to specialised neurological care is limited.

Traditional stroke-risk assessment relies on clinical scores such as the Framingham Stroke Risk Score and $CHA_2DS_2$-VASc. These scores do not capture complex, non-linear interactions among risk factors and do not use dynamic physiological signals [4]. Electromyography (EMG), which records the electrical activity of skeletal muscle, is a promising complementary modality: stroke-induced motor impairment alters EMG amplitude, frequency content, inter-muscle coordination and activation timing, and such alterations may be measurable before overt clinical signs appear [5, 6]. Surface EMG (sEMG) is non-invasive, inexpensive and portable, which makes it attractive for primary-care screening where Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) are unavailable.

Recurrent Neural Networks (RNNs) are designed for sequential data and have shown promise in classifying biomedical time series [7]. Their training, however, is susceptible to vanishing and exploding gradients [8], which techniques such as gradient clipping and learning-rate scheduling are known to mitigate [9]. This paper presents an integrated RNN-based architecture for automated stroke detection from EMG data. Its specific contributions are:

(1) A concise review of AI/ML architectures for stroke detection, with emphasis on time-series modalities such as EMG.
(2) An integrated RNN-based architecture that combines EMG preprocessing, sliding-window segmentation, subject-wise data partitioning, class-weighted training, gradient clipping and learning-rate scheduling with GRU and CNN-GRU temporal encoders.
(3) A scenario-based evaluation of the architecture with ATAM, extended to eight quality-attribute scenarios organised in a utility tree.
(4) An analysis of risks, trade-offs and vulnerabilities, the impact of architectural changes on performance criteria, and a validation protocol for future empirical work.

The remainder of the paper is organised as follows. Section 2 reviews related concepts and literature. Section 3 analyses an existing architecture and presents the proposed one. Section 4 describes the scenario-based evaluation, Section 5 discusses the results and limitations, and Section 6 concludes the paper.

## 2. RELATED CONCEPTS AND LITERATURE
### 2.1 Stroke pathophysiology and EMG
A stroke occurs when blood supply to part of the brain is interrupted or reduced, depriving neural tissue of oxygen and

nutrients [10]. Ischaemic stroke, caused by a thrombus or embolus blocking an artery, accounts for approximately 85% of cases. Haemorrhagic stroke, caused by the rupture of a vessel within the brain (intracerebral) or in the surrounding space (subarachnoid), accounts for about 15% of cases but a disproportionate share of deaths. A Transient Ischaemic Attack (TIA) is a temporary blockage whose symptoms resolve quickly; it is an important warning sign because a substantial proportion of untreated patients go on to have a major stroke [10]. Common symptoms include sudden unilateral weakness or numbness of the face, arm or leg, speech disturbance, visual loss and loss of balance.

The EMG signal is the summation of motor unit action potentials recorded from a muscle [6]. Features relevant to stroke include amplitude (related to motor-unit recruitment, often reduced in paresis), frequency content (related to motor-unit firing rates), timing and coordination between muscles (abnormal co-contraction and delayed activation are common after stroke), signal complexity, and fatigue characteristics. Figure 1 relates the stroke types to their common pathway and to the sEMG alterations that motivate the proposed architecture.

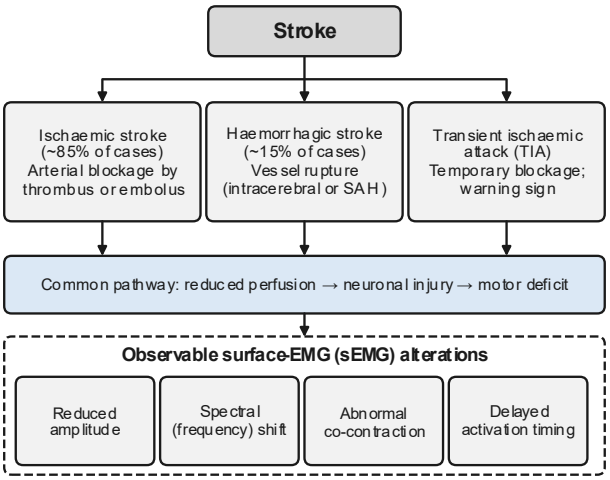



**Fig 1: Stroke types, common pathway and observable sEMG alterations**

## 2.2 Machine learning in stroke detection

Machine learning (ML) learns predictive models from data and applies them to unseen cases; deep learning (DL) uses multi-layer neural networks that learn features and classifiers jointly from raw inputs, avoiding manual feature engineering [11, 12]. Figure 2 summarises the main learning paradigms. Supervised learning is the paradigm used here, since EMG recordings are labelled as stroke or healthy; semi-supervised learning is relevant because labelled clinical data are scarce. Small datasets cause overfitting, and pooling data from several sites is difficult because of heterogeneous protocols and devices [13]; these issues are central to the architecture of any AI-based diagnostic system. Automated analysis pipelines in medical imaging typically follow segmentation, feature extraction, feature selection and classification [14], a sequence that also applies to EMG analysis.

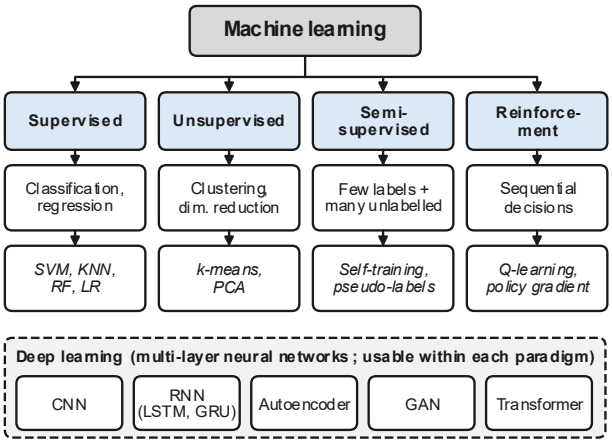



**Fig 2: Taxonomy of machine learning approaches**

## 2.3 Review of AI/ML architectures for stroke detection

ML approaches have been used from early identification of stroke to prognosis after therapy [15]. Lakkshmanan et al. [16] combined Logistic Regression (LR), Random Forest (RF), XGBoost, AdaBoost and GRU classifiers with an Adaptive Neuro-Fuzzy Inference System (ANFIS), which performed best on their data. Such systems depend on domain-specific feature engineering, are sensitive to missing values and outliers, and generalise poorly to data acquired with other devices or in other settings. Deep learning removes manual feature engineering but requires larger datasets. Guntari et al. [7] combined a genetic algorithm with an RNN to classify post-stroke EEG signals with 90% test accuracy, and Yu et al. [5] reported 98.58% test accuracy with an LSTM on real-time EMG, although performance decreased on real-world data. Bidirectional RNNs [17] and bidirectional LSTMs process sequences forwards and backwards and outperform unidirectional networks in tasks such as phoneme classification [18].

Table 1 summarises the architectural limitations of existing systems identified in the review and the corresponding responses in the proposed architecture; together they define the research gap addressed in this paper.

**Table 1. Limitations of existing systems and architectural responses**

| Limitation | Response in proposed architecture |
|---|---|
| No temporal modelling of EMG time series | GRU / CNN-GRU encoders over sliding windows |
| Unstable training of recurrent models | Gradient clipping and LR schedulers |
| Sensitivity to noise, missing data, outliers | Filtering, DC removal, normalisation, data-quality checks |
| Bias towards the majority (healthy) class | Class-weighted loss and subject-wise splits |
| Limited interpretability | Interactive visual explanation module |
| Poor scalability, high compute cost | Layered, modular, distribution-ready design |
| Weak integration with clinical systems | Data access layer with EHR interface |

# 3. PROPOSED RNN-BASED ARCHITECTURE

## 3.1 Analysis of existing architectures

Two existing architectures were analysed. Figure 3 shows a typical Convolutional Neural Network (CNN), with input, convolution, pooling, fully connected and output layers. CNNs extract spatial features effectively but do not, on their own, model long-range temporal dependencies in sequential signals.

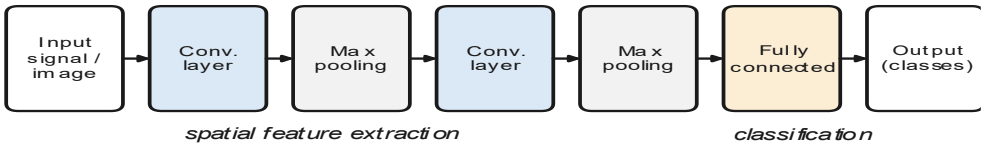



**Fig 3: Typical CNN architecture**

Figure 4 shows the stroke prediction system of Lakkshmanan et al. [16]. It acquires the Kaggle stroke prediction dataset (clinical records) and MRI images, preprocesses them (redundancy removal, label encoding, mean imputation of missing values, data balancing and outlier rejection), applies SegNet-based segmentation and Capsule Network (CapsNet) feature extraction, and compares LR (baseline), RF, XGBoost, AdaBoost, GRU and ANFIS classifiers.

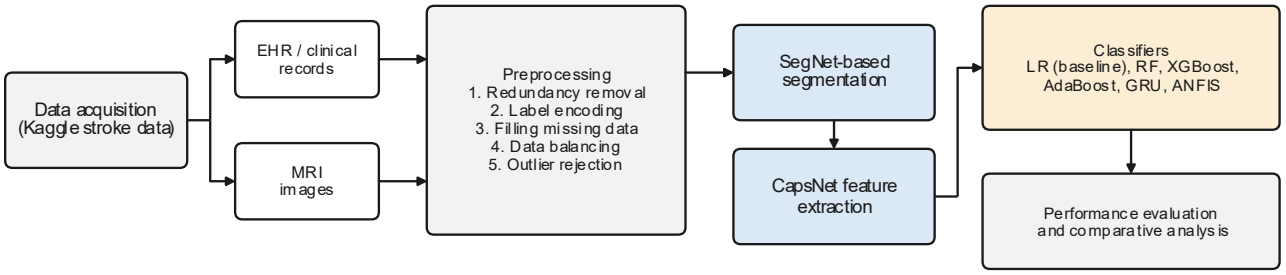



**Fig 4: Architecture of the existing system (adapted from Lakkshmanan et al. [16])**

The analysis identified the following shortcomings: low accuracy of the baseline LR model; sensitivity to missing data, noise and outliers; no processing of time-series signals such as EMG; high computational cost (particularly ANFIS) with weak generalisation on the test set; no design for large-scale screening; limited multimodal integration; and limited explanation of predictions.

## 3.2 Recurrent neural networks

An RNN shares parameters across time steps, computing a hidden state $h_t$ from the current input $x_t$ and the previous state $h_{t-1}$, and an output $y_t$ from the hidden state (Figure 5):

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (1)$$

$$y_t = g(W_{hy}h_t + b_y) \quad (2)$$

where $f(\cdot)$ and $g(\cdot)$ are activation functions. Repeated multiplication by $W_{hh}$ during back-propagation through time causes gradients to vanish or explode [8], which motivated gated units.

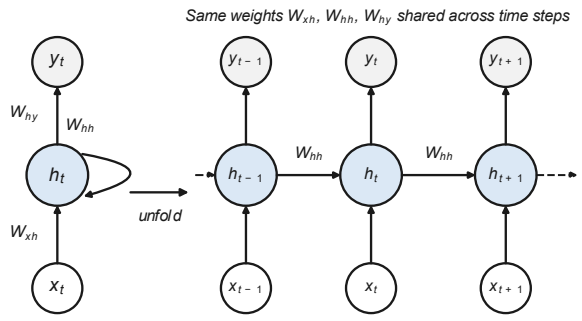



**Fig 5: Folded and unfolded recurrent neural network**

The Gated Recurrent Unit (GRU) [19] uses a reset gate $r_t$ and an update gate $z_t$:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (3)$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (4)$$

$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \quad (5)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (6)$$

where $\sigma$ is the logistic sigmoid and $\odot$ denotes element-wise multiplication. The Long Short-Term Memory (LSTM) unit [20] adds a cell state $c_t$ controlled by forget ($f_t$), input ($i_t$) and output ($o_t$) gates:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (7)$$

$$h_t = o_t \odot \tanh(c_t) \quad (8)$$

For stroke detection, recurrent models offer temporal modelling of muscle activity, contextual learning across time steps, end-to-end learning from minimally processed signals and tolerance of variable recording lengths.

## 3.3 Integrated RNN-based architecture

The proposed architecture (Figure 6) is designed for the one-dimensional sequential structure of EMG. Unlike the existing system, it does not retain the SegNet and CapsNet components, which were designed for two-dimensional images and whose applicability to windowed 1-D EMG was not established; feature extraction and temporal modelling are instead performed by the recurrent layers. The pipeline has five stages, supported by a training-control block and an output and deployment block.

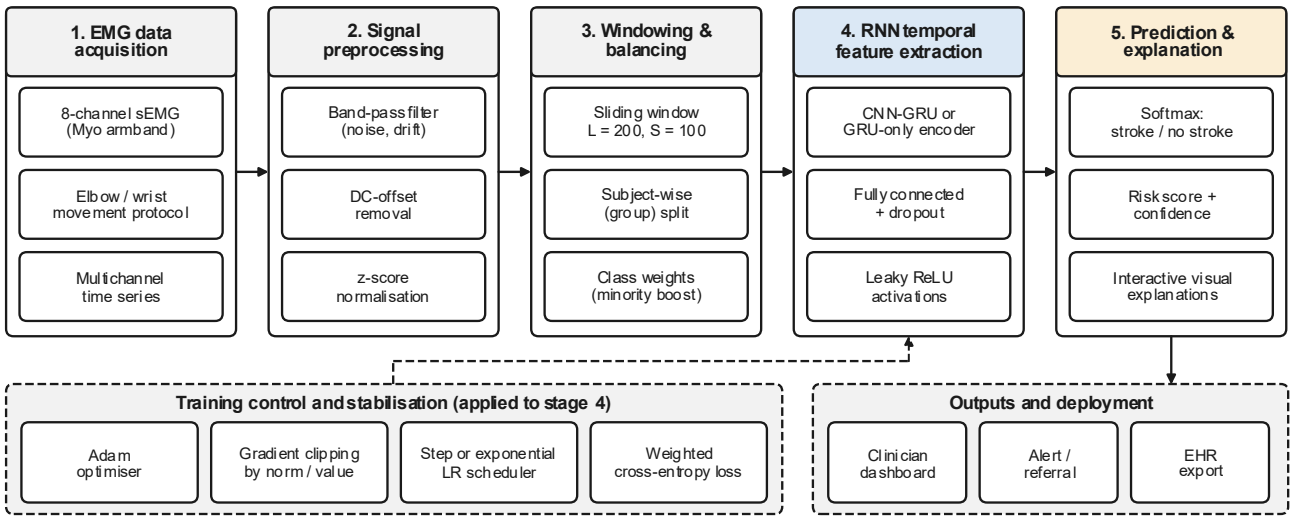



**Fig 6: Proposed RNN-based architecture for stroke detection and classification**

**Stage 1 – Input.** The MUSED-I sEMG dataset [21] is used to inform the design of the input and preprocessing layers. It contains recordings from 11 healthy subjects and 2 stroke patients captured with an 8-channel Myo armband during a dynamic arm-movement protocol implemented in Python with the Pygame library. Table 2 summarises the dataset and the pipeline parameters. Because the stroke cohort contains only two subjects, the dataset is not used, and is not statistically sufficient, for clinical performance evaluation (Section 5.5).

**Table 2. Input data and pipeline parameters**

| Item | Value |
|---|---|
| Dataset | MUSED-I sEMG [21] |
| Subjects | 11 healthy, 2 stroke |
| Device / channels | Myo armband, 8 sEMG channels |
| Window length $L$ / step $S$ | 200 / 100 samples (50% overlap) |
| Normalisation | Per-channel z-score, $\epsilon = 10^{-6}$ |
| Output classes | Stroke, no stroke |
| Data partitioning | Subject-wise (grouped) folds |

**Stage 2 – Preprocessing.** Signals are converted to 32-bit floating point, band-pass filtered to suppress high-frequency noise and low-frequency drift [6], DC offset is removed and each channel $c$ is z-score normalised:

$$\tilde{x}_{t,c} = \frac{x_{t,c} - \mu_c}{\sigma_c + \epsilon} \tag{9}$$

where $\mu_c$ and $\sigma_c$ are the channel mean and standard deviation. Redundant records are removed and labels are encoded numerically.

**Stage 3 – Windowing and balancing.** A recording of $T$ samples is segmented into overlapping windows of length $L$ with step $S$, giving

$$N_w = \lfloor (T - L)/S \rfloor + 1 \tag{10}$$

windows. Windows inherit the subject label, and all windows of a subject are kept in the same fold (group-based cross-validation), so that overlapping windows of one person never appear in both training and test sets. Class imbalance is handled with weights inversely proportional to the class counts $n_0$ (healthy) and $n_1$ (stroke), normalised so that they sum to two:

$$w_k = \frac{2/n_k}{1/n_0 + 1/n_1}, \quad k \in \{0,1\} \tag{11}$$

These weights enter the weighted cross-entropy loss over $N$ training windows with predicted class probabilities $\hat{p}$:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} w_{y_i} \log \hat{p}_{i,y_i} \tag{12}$$

**Stage 4 – Recurrent temporal feature extraction.** Two candidate encoders are specified for comparative evaluation: a CNN-GRU model, in which a 1-D convolution extracts local features before a GRU captures longer-term dependencies, and a simpler GRU-only model. Fully connected layers with dropout [22] follow the encoder. Leaky ReLU [23] avoids inactive ("dead") neurons:

$$\text{LeakyReLU}(x) = \max(0.01x,\ x) \tag{13}$$

**Stage 5 – Prediction and explanation.** The output layer produces a probability distribution over the two classes with the softmax function:

$$\text{softmax}(\mathbf{x})_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}} \tag{14}$$

The predicted class, a risk score and a confidence value are passed to an interactive visual explanation component that lets clinicians inspect the input signals and the model output.

**Training control.** Parameters are optimised with Adam [24]. Gradient clipping by norm rescales the gradient vector $\mathbf{g}$ when its norm exceeds a threshold $\tau$ (e.g. 1.0) [9], and clipping by value bounds each component to $[-c, c]$:

$$\mathbf{g} \leftarrow \mathbf{g} \cdot \min\left(1,\ \frac{\tau}{\|\mathbf{g}\|_2}\right) \tag{15}$$

$$g_j \leftarrow \min\left(\max\left(g_j,\ -c\right),\ c\right) \tag{16}$$

Step decay and exponential decay schedulers adjust the learning rate $\eta$ at epoch $e$, allowing large updates early and fine-tuning later, where $\eta_0$ is the initial rate, $d$ the drop factor, $s$ the step size and $k$ the decay constant:

$$\eta_e = \eta_0 \cdot d^{\lfloor e/s \rfloor} \tag{17}$$

$$\eta_e = \eta_0 \cdot e^{-ke} \tag{18}$$

## 3.4 System architecture layers

The system follows a three-layer design (Figure 7). The presentation layer provides the clinician dashboard, signal plots,

risk reports and explanations. The business logic layer implements acquisition, preprocessing, model inference and decision rules. The data access layer stores patient records, EMG recordings, versioned model parameters and audit logs, and exchanges data with external EHR and hospital information systems. Security, access control and logging cut across all layers.

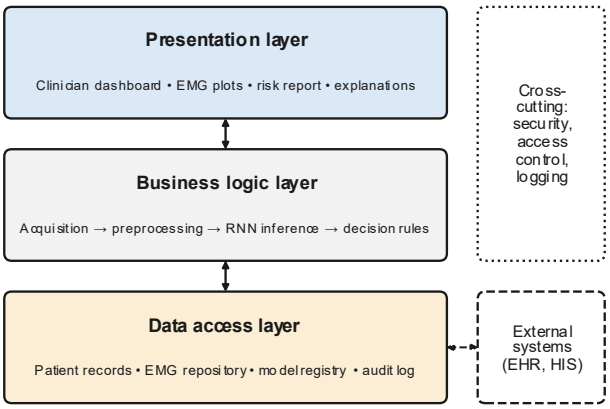



**Fig 7: System architecture layers of the proposed system**

## 3.5 Architectural design decisions

**Modularity:** each module addresses one concern, which eases maintenance, testing and replacement of the model. **Scalability:** stateless inference services allow horizontal scaling for screening campaigns. **Reliability:** critical components are redundant and failures are handled explicitly. **Interoperability:** the data access layer is designed to integrate with EHR and hospital information systems. **Compliance:** regulatory compliance is a forward-looking design consideration; depending on the deployment jurisdiction it may involve HIPAA, the GDPR or a national regulation such as Nigeria's NDPR. No regulatory clearance or Software-as-a-Medical-Device classification has been sought at this design stage.

# 4. SCENARIO-BASED EVALUATION

Architecture evaluation can take place at any stage of development: early in design to compare options, or later to assess existing systems and detect architectural drift. Evaluation methods are commonly grouped into experience-based, simulation-based, mathematical-modelling-based and scenario-based methods [25]. Scenario-based methods are suited to architectures that have not yet been implemented: they describe interactions between the system and its environment (user actions, failures, attacks or changes) and examine how the architecture responds [26]. This study applies the Architecture Trade-off Analysis Method (ATAM) [26, 27], which evaluates an architecture against several quality attributes and exposes the trade-offs among them.

## 4.1 Primary scenario: stroke-risk screening in a clinic

**Description.** A patient with stroke risk factors (e.g. hypertension, diabetes, smoking, age over 55) attends a primary-care clinic for a routine check-up, which triggers the automated stroke-risk assessment. This scenario represents resource-limited settings with restricted access to neurologists and imaging.

**User action.** A nurse fits a portable 8-channel EMG device to the forearm and upper arm, and the patient performs elbow flexion and extension and wrist flexion and extension. The recording is uploaded to the detection system.

**System response.** The system preprocesses and windows the signals and applies the GRU-based model to detect patterns indicative of stroke-related neuromuscular abnormality, such as reduced amplitude, altered frequency content and abnormal co-contraction. A high-risk result generates an alert recommending neurological consultation; a low-risk result produces a report with signal plots and the model's confidence score.

**Evaluation criteria.** Detection accuracy; false-positive and false-negative rates; computational efficiency; scalability; robustness to noise, artefacts and demographic variation; interpretability; reliability; and security of patient data.

## 4.2 Application of ATAM

Table 3 shows how the ATAM steps were applied to the primary scenario. The evaluation, including the stakeholder perspectives in the evaluation-meeting step, was conducted as an author-led architectural walkthrough consistent with a design-stage application of ATAM. It represents anticipated stakeholder concerns informed by the clinical framing in Section 2, not an empirical workshop with clinicians, patients or administrators; such a workshop is identified as future work.

**Table 3. Application of ATAM to the stroke-detection scenario**

| ATAM step | Description | Application to the scenario |
|---|---|---|
| Preparation | Define stakeholders, goals, constraints and scenarios. | Stakeholders: patients, doctors, nurses, system and healthcare administrators, researchers, regulators. Goals: sensitivity > 80%, < 5 min per analysis, > 1,000 screenings per day, robustness to noise. Scenarios S1–S8 (Figure 8). |
| Evaluation meeting | Present the architecture and scenarios; elicit concerns. | Alignment with clinical requirements for accuracy, speed and reliability. Clinicians are expected to prioritise sensitivity over specificity; administrators, cost and throughput. |
| Identify architectural approaches | Explore alternatives that address the concerns. | GRU-only, CNN-GRU and LSTM encoders; quantisation and pruning; cloud versus on-premises or on-device deployment; registry-based model management. |
| Analyse trade-offs | Analyse effects of approaches on quality attributes. | Model complexity versus interpretability; accuracy versus computational cost; cloud scalability versus connectivity and privacy; sensitivity versus specificity. |
| Documentation and decision-making | Record findings, decisions and recommendations. | Adopt the GRU-only model as the default for its balance of accuracy and efficiency; keep CNN-GRU as an alternative; roadmap of dataset expansion and model refinement. |

## 4.3 Extended scenario set and utility tree

To extend the evaluation beyond a single use case, the quality-attribute goals were refined into eight concrete scenarios and organised as an ATAM utility tree (Figure 8). Each scenario is ranked by its importance to stakeholders and its difficulty for the architecture. The scenarios cover safety-critical detection (S1, S2), high-volume screening (S3), signal faults (S4), offline

operation (S5), clinical-record integration (S6), model evolution (S7) and accessibility for patients who cannot complete the protocol (S8).

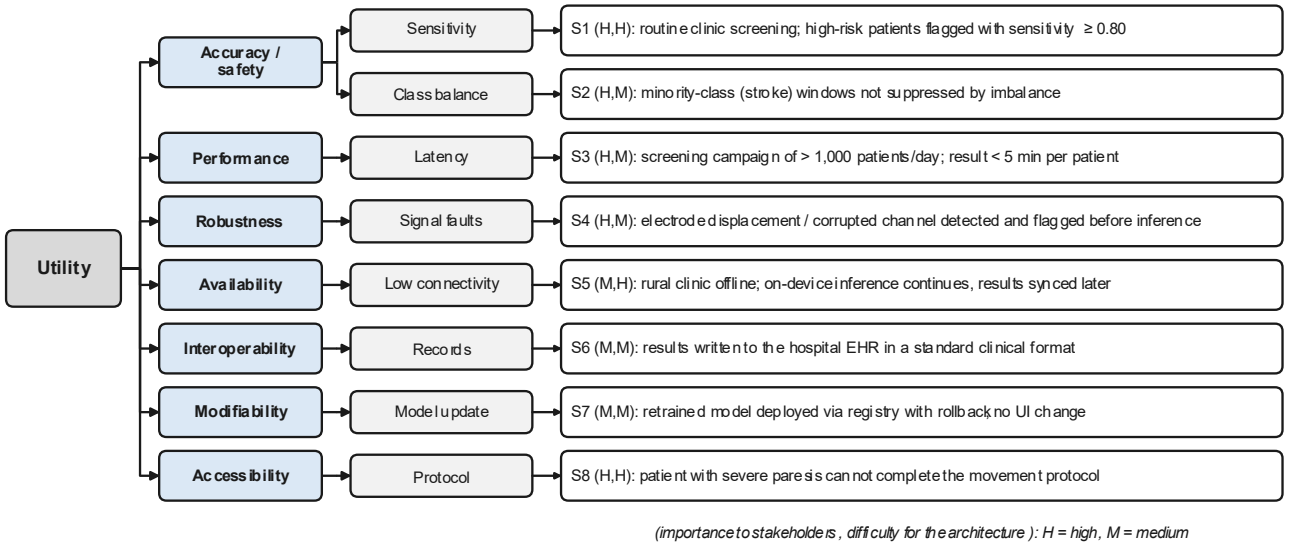



**Fig 8: ATAM utility tree with eight quality-attribute scenarios**

Table 4 analyses each scenario against the architectural decisions that support it and records the resulting sensitivity points, trade-off points, risks and non-risks.

**Table 4. Scenario analysis: architectural decisions, sensitivity and trade-off points, risks and non-risks**

| Scenario | Supporting decisions | Sensitivity / trade-off points | Risk (R) / non-risk (N) |
|---|---|---|---|
| S1 Clinic screening | GRU encoder; softmax risk score; adjustable threshold | Decision threshold trades sensitivity against specificity | R: two-subject stroke cohort cannot establish sensitivity |
| S2 Class imbalance | Class weights (Eq. 11); weighted loss (Eq. 12); subject-wise folds | Weight magnitude trades recall of stroke class against false alarms | N: imbalance handled explicitly in the training stage |
| S3 Screening campaign | Stateless inference services; asynchronous queue; result caching | Model size and batch size drive latency and cost | N: windowed GRU inference is lightweight relative to imaging pipelines |
| S4 Corrupted channel | Filtering; per-channel quality checks before inference | Strict quality thresholds raise repeat-recording rate | R: electrode placement variability across operators not yet modelled |
| S5 Offline rural clinic | On-device quantised model; deferred synchronisation | Quantisation trades accuracy for footprint | R: local storage of patient data increases security exposure |
| S6 EHR integration | Data access layer with standard clinical interface | Interface standard choice affects portability | N: integration isolated in one layer |
| S7 Model update | Model registry with versioning and rollback | Retraining cadence trades freshness against validation effort | R: each update requires re-validation for regulatory traceability |
| S8 Severe paresis | Protocol is a configurable input; operator can mark partial recordings | Adapted protocols change the input distribution | R: protocol adaptations for severely impaired patients not yet designed |

## 4.4 Validation protocol for future empirical work

Because no model was trained in this study, a protocol is specified so that later empirical results can be compared fairly. Performance should be estimated with leave-one-subject-out or grouped k-fold cross-validation, reporting accuracy, sensitivity, specificity, F1-score and area under the ROC curve at subject level as well as window level. Before clinical claims are made, the stroke cohort should be expanded to at least 20–30 patients. Larger public sEMG corpora such as NinaPro [28] can be used to pre-train the encoder before fine-tuning on stroke data, and robustness should be tested with injected noise and simulated channel loss to exercise scenario S4.

# 5. RESULTS AND DISCUSSION

This section reports the findings of the scenario-based evaluation. They are qualitative architectural findings, not empirical performance results: no model was trained or tested. The architecture is designed to improve detection accuracy, reduce false positives, scale to screening workloads and tolerate noisy data; whether it achieves these goals remains to be demonstrated (Sections 4.4 and 5.5).

## 5.1 Architectural strengths

- **Temporal modelling:** recurrent layers capture dependencies across EMG samples, which existing systems treating time points independently cannot.
- **Training stability:** gradient clipping and learning-rate schedulers address vanishing and exploding gradients.
- **Modularity and interoperability:** the layered design separates concerns and isolates EHR integration.
- **Interpretability:** the visual explanation component supports clinician trust.
- **Scalability and compliance readiness:** stateless inference and audit logging support distribution and traceability.

## 5.2 Risks and trade-offs

**Overfitting (risk).** With only two stroke patients, the model could memorise training data instead of learning generalisable features. Dropout, batch normalisation and data augmentation can mitigate this, but the limitation is treated as a boundary condition: no accuracy or generalisation claim is made until evaluation on a larger cohort.

**Accuracy versus computational cost (trade-off).** Deeper or bidirectional CNN-GRU models may be more accurate but are less suited to resource-constrained clinics; the architecture therefore supports pruning and quantisation.

**Complexity versus interpretability (trade-off).** More complex models are harder to explain. The GRU-only model offers a better balance, and the explanation component partly offsets the black-box nature of the model.

**Dataset imbalance and data quality (vulnerabilities).** A minority stroke class biases the model towards healthy predictions, raising false negatives; class weighting and synthetic data generation are required. Inaccurate or biased recordings can mislead the model, so data-quality monitoring is needed.

**Regulation and deployment (risk and trade-off).** Regulatory approval is lengthy and costly, requiring traceable predictions and audit trails. Cloud deployment offers scalability and easy updates but needs reliable connectivity and raises privacy concerns; on-premises deployment improves data control at a higher infrastructure cost.

## 5.3 Impact of changes on performance criteria

Table 5 summarises architectural changes and the performance criteria they are expected to influence.

**Table 5. Expected impact of changes on performance criteria**

| Criterion | Candidate changes |
|---|---|
| Detection accuracy | Larger stroke cohort; augmentation (time warping, noise injection, shifting); hyperparameter tuning; bidirectional LSTM or attention |
| False positives / negatives | Threshold optimisation; cost-sensitive learning; minority-class weighting |
| Computational efficiency | Pruning; quantisation; depthwise-separable convolutions; GPU or edge accelerators |
| Scalability | Load balancing and horizontal scaling; result caching; asynchronous pipelines |
| Robustness | Data validation; adversarial training; model ensembles |
| Interpretability | Attention over time; SHAP [29] or LIME; rule extraction |

## 5.4 Feedback and iterative refinement

The evaluation shows that detection accuracy, computational efficiency and robustness must be balanced. The main feedback is the need for a larger and more diverse dataset, better handling of class imbalance and stronger interpretability. Recommended refinements are: (i) data-quality monitoring in the pipeline; (ii) a model registry with versioning; (iii) continuous monitoring of model performance in production; (iv) a clinician feedback loop; and (v) A/B testing of model updates.

## 5.5 Limitations

The following limitations are boundary conditions on the claims of this paper. **(a) No empirical validation:** the architecture was not trained or tested on any dataset in this study, so quantitative performance claims are out of scope. **(b) Sample size:** the stroke cohort (n = 2) is far below the 20–30 patients considered necessary for generalisable claims. **(c) Protocol accessibility:** the movement protocol assumes enough motor function to perform elbow and wrist movements; patients with severe impairment may be unable to complete it (scenario S8). **(d) Operator-dependent variability:** sEMG quality depends on consistent electrode placement across operators and sessions, which is not yet addressed. **(e) Regulatory scope:** the regulatory considerations in Section 3.5 are prospective, not a demonstrated compliance status.

# 6. CONCLUSION

This paper presented an integrated RNN-based architectural framework for the automated detection and diagnosis of stroke from EMG data, together with a scenario-based evaluation using ATAM. The architecture combines EMG preprocessing, sliding-window segmentation, subject-wise partitioning and class-weighted training with GRU and CNN-GRU encoders stabilised by gradient clipping and learning-rate scheduling. The evaluation, extended to eight quality-attribute scenarios, identified the sensitivity points, trade-offs, risks and non-risks of the design and the anticipated impact of changes on accuracy, efficiency, scalability, robustness and interpretability. These findings are architectural and have not yet been validated empirically.

Future work should: (1) train and test the architecture under the validation protocol of Section 4.4, which is the precondition for any clinical claim; (2) expand the dataset through multi-institution collaboration; (3) investigate Transformer and attention-based encoders; (4) integrate patient history and other signals such as EEG and accelerometry; (5) improve explanations and report confidence intervals; (6) develop robust preprocessing for real-world artefacts; and (7) prepare for regulatory approval and for cloud and on-premises deployment with continuous monitoring.

# 7. REFERENCES

[1] Xia, X., Yue, W., Chao, B., Li, M., Cao, L., Wang, L., Shen, Y. and Li, X. 2019 Prevalence and risk factors of stroke in the elderly in Northern China: Data from the National Stroke Screening Survey. Journal of Neurology. 266(6): 1449-1458.

[2] Lindsay, M.P., Norrving, B., Sacco, R.L., Brainin, M., Hacke, W., Martins, S., Pandian, J. and Feigin, V. 2019 World Stroke Organization (WSO): Global stroke fact sheet 2019. International Journal of Stroke. 14(8): 806-817.

[3] World Health Organization 2019 Global Health Estimates: Life expectancy and leading causes of death and disability. World Health Organization, Geneva.

[4] Adam, S.Y., Yousif, A. and Bashir, M.B. 2016 Classification of ischemic stroke using machine learning algorithms. International Journal of Computer Applications. 149(10): 26-31.

[5] Yu, J., Park, S., Kwon, S.H., Ho, C.M.B., Pyo, C.S. and Lee, H. 2020 AI-based stroke disease prediction system using real-time electromyography signals. Applied Sciences. 10(19): 6791.

[6] De Luca, C.J. 1997 The use of surface electromyography in biomechanics. Journal of Applied Biomechanics. 13(2): 135-163.

[7] Guntari, E.W., Djamal, E.C., Nugraha, F. and Liem, S.L.L. 2020 Classification of post-stroke EEG signal using genetic algorithm and recurrent neural networks. Proceedings of the 7th International Conference on Electrical Engineering, Computer Sciences and Informatics (EECSI). 156-161.

[8] Bengio, Y., Simard, P. and Frasconi, P. 1994 Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks. 5(2): 157-166.

[9] Pascanu, R., Mikolov, T. and Bengio, Y. 2013 On the difficulty of training recurrent neural networks. Proceedings of the 30th International Conference on Machine Learning (ICML). 1310-1318.

[10] Tang, H., Shen, H. and Qi, H. 2011 Stroke: A review of the literature. Journal of Biomedical Research. 25(3): 157-168.

[11] Cuocolo, R., Cipullo, M.B., Stanzione, A., Ugga, L., Romeo, V., Radice, L., Brunetti, A. and Imbriaco, M. 2019 Machine learning applications in prostate cancer magnetic resonance imaging. European Radiology Experimental. 3: 35.

[12] Laudicella, R., Comelli, A., Stefano, A., Szostek, E., Cammaroto, G., Baldari, S. et al. 2021 Artificial neural networks in cardiovascular diseases and its potential for clinical application in molecular imaging. Current Radiopharmaceuticals. 14(3): 209-219.

[13] Foster, B., Bagci, U., Mansoor, A., Xu, Z. and Mollura, D.J. 2014 A review on segmentation of positron emission tomography images. Computers in Biology and Medicine. 50: 76-96.

[14] Szczypiński, P.M., Strzelecki, M., Materka, A. and Klepaczko, A. 2009 MaZda – A software package for image texture analysis. Computer Methods and Programs in Biomedicine. 94(1): 66-76.

[15] Kourou, K., Exarchos, T.P., Exarchos, K.P., Karamouzis, M.V. and Fotiadis, D.I. 2015 Machine learning applications in cancer prognosis and prediction. Computational and Structural Biotechnology Journal. 13: 8-17.

[16] Lakkshmanan, A., Suji, A., Priyanka, N., Anand, D.B., Nagaraju, K. and Naidu, U.U. 2023 A novel machine learning based stroke prediction system using magnetic resonance image and adaptive neuro-fuzzy inference. International Journal of Intelligent Systems and Applications in Engineering. 12(10): 576-585.

[17] Schuster, M. and Paliwal, K.K. 1997 Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing. 45(11): 2673-2681.

[18] Graves, A. and Schmidhuber, J. 2005 Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks. 18(5-6): 602-610.

[19] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y. 2014 Learning phrase representations using RNN encoder-decoder for statistical machine translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 1724-1734.

[20] Hochreiter, S. and Schmidhuber, J. 1997 Long short-term memory. Neural Computation. 9(8): 1735-1780.

[21] Marwat, M. and Human System Lab, SMME, NUST Islamabad 2023 MUSED-I: Surface electromyography (sEMG) dataset for stroke rehabilitation [Data set]. Kaggle. Retrieved from: https://www.kaggle.com/datasets/mustafamarwat/mused-i-semg-dataset-for-stroke-rehabilitation

[22] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. 2014 Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research. 15(56): 1929-1958.

[23] Maas, A.L., Hannun, A.Y. and Ng, A.Y. 2013 Rectifier nonlinearities improve neural network acoustic models. Proceedings of the ICML Workshop on Deep Learning for Audio, Speech and Language Processing. Atlanta, USA.

[24] Kingma, D.P. and Ba, J. 2015 Adam: A method for stochastic optimization. Proceedings of the 3rd International Conference on Learning Representations (ICLR). San Diego, USA.

[25] Dobrica, L. and Niemelä, E. 2002 A survey on software architecture analysis methods. IEEE Transactions on Software Engineering. 28(7): 638-653.

[26] Bass, L., Clements, P. and Kazman, R. 2012 Software Architecture in Practice, 3rd ed. Addison-Wesley, Boston.

[27] Kazman, R., Klein, M. and Clements, P. 2000 ATAM: Method for architecture evaluation. Technical Report CMU/SEI-2000-TR-004. Software Engineering Institute, Carnegie Mellon University, Pittsburgh.

[28] Atzori, M., Gijsberts, A., Castellini, C., Caputo, B., Hager, A.G.M., Elsig, S., Giatsidis, G., Bassetto, F. and Müller, H. 2014 Electromyography data for non-invasive naturally-controlled robotic hand prostheses. Scientific Data. 1: 140053.

[29] Lundberg, S.M. and Lee, S.-I. 2017 A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems (NeurIPS). 30: 4765-4774.