Embedded Trustworthy AI for Healthcare

A Multi-Objective Study of Fairness, Privacy, and Efficiency under TinyML Constraints

Luca Tompea[1]

Supervisor: Dr. Qing Wang[1]

[1]**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 21, 2026

Name of the student: Luca Tompea
Final project course: CSE3000 Research Project
Thesis committee: Dr. Qing Wang, Prof. Dr. Mark Neerincx

Abstract

The growing deployment of AI-assisted diagnostics on resource-constrained microcontrollers raises an underexplored question: do the memory and latency limitations of embedded hardware reshape the fairness–accuracy–privacy trade-offs that practitioners must navigate in healthcare applications? We present a controlled, multi-objective empirical study evaluating Gaussian noise injection, post-training INT8 quantization, and classification threshold calibration. Fairness and privacy interventions are evaluated on the Pima Indians Diabetes dataset (768 samples, age-stratified protected group) using a lightweight MLP and a logistic regression baseline; quantization efficiency is additionally validated on a larger hospital readmission dataset ($\sim$100,000 samples, $\sim$154,800-parameter model) to characterise scale-dependent compression behaviour. The key findings are fourfold: (i) INT8 quantization efficiency is scale-dependent: no benefit and up to 67% fairness degradation at sub-300 parameters, versus 3.87$\times$ compression and 3.6$\times$ speedup at $\sim$155k parameters; (ii) low-magnitude noise (σ=0.05) is a safe privacy proxy with negligible accuracy cost; (iii) higher noise levels create a non-monotonic privacy–fairness tension, destabilising group-level fairness without predictably improving it; (iv) post-hoc threshold calibration to τ=0.7 reduces equalized odds gap by 18.4% relative at only 1.2 pp accuracy cost, outperforming all training-time interventions with zero embedded overhead. These findings show that embedded constraints do not introduce new fairness–accuracy trade-offs but shift design priorities toward post-deployment calibration.

1 Introduction

Algorithmic bias in medical AI is not a theoretical concern but an observable, harmful reality [1]. As AI-assisted diagnostics increasingly migrate from hospital servers to wearable monitors and point-of-care devices, a new dimension of risk emerges: the resource constraints of embedded hardware may interact with model fairness and privacy in ways that practitioners have not yet characterised.

TinyML, the deployment of trained models on microcontrollers (MCUs) with kilobyte-scale memory and milliwatt power budgets, has emerged as an enabling paradigm for autonomous, low-latency medical inference without cloud connectivity [2, 3]. Benchmarks such as MLPerf Tiny have driven rapid progress in model compression and efficient inference [3], yet the dominant optimisation target in this literature remains predictive accuracy under hardware constraints. The trustworthy AI dimensions that are critical in healthcare, fairness and privacy, have received comparatively little attention in the embedded setting. It remains unknown whether compression techniques such as INT8 quantization alter group-level fairness, and whether noise magnitudes suitable as privacy proxies are compatible with stable fairness metrics under hardware constraints.

This paper addresses that gap through a controlled, multi-objective empirical study. The central research question is:

> *How do model quantization and privacy-preserving input perturbations influence the trade-off between predictive accuracy and group fairness in healthcare TinyML models?*

To investigate this question, we conduct experiments on two healthcare datasets. Fairness and privacy interventions are evaluated on the Pima Indians Diabetes dataset [4] under a factorial design crossing five noise levels and five classification thresholds, with an age-stratified protected group defined by the dataset median and both a lightweight MLP and a logistic regression baseline. To characterise the scale-dependence of quantization efficiency, we additionally train a larger MLP on the Diabetes 130-US Hospitals dataset [5] ($\sim$101,000 samples), converting it to both FP32 and INT8 TFLite and benchmarking size and inference latency on a Raspberry Pi 4.

The main contributions of this work are as follows:

1. A reproducible experimental framework that jointly evaluates fairness, privacy, and embedded efficiency under controlled deployment conditions for healthcare classification.
2. A scale-dependent characterisation of INT8 quantization: at the sub-300-parameter scale of the Pima model, INT8 provides no size or latency benefit and amplifies fairness disparities by up to 67% under noise; at the $\sim$155,000-parameter scale of the hospital readmission model, INT8 achieves 3.87$\times$ size reduction and 3.6$\times$ latency speedup on embedded hardware.
3. Characterisation of a non-monotonic privacy–fairness tension: low-magnitude Gaussian noise injection is a safe privacy proxy, but higher levels destabilise group-level fairness metrics in ways that are not predictable from accuracy alone.
4. Demonstration that post-hoc classification threshold calibration is the most effective low-cost fairness intervention in this setting, giving an 18.4% relative reduction in equal opportunity gap at a cost of 1.2 pp in accuracy with no embedded overhead.

The remainder of the paper is structured as follows. Section 2 reviews related work on trustworthy AI in healthcare and TinyML. Section 3 describes the datasets, experimental design, model architectures, embedded deployment setup, and evaluation metrics. Section 4 presents the experimental results across all configurations. Section 5 discusses responsible research considerations. Section 6 interprets the findings and discusses limitations. Section 7 concludes and outlines future work.

2 Related Work and Background

This section reviews fairness and privacy in healthcare AI, embedded machine learning and TinyML, and identifies the research gap this study addresses.

2.1 Trustworthy AI in Healthcare

Algorithmic bias in clinical decision support is not a theoretical concern but an observed, consequential problem. Obermeyer et al. demonstrated that a widely deployed commercial risk-management algorithm systematically assigned lower risk scores to Black patients than to equally ill White patients, resulting in significantly fewer referrals to specialist care [1]. This work established that disparate predictive error rates across demographic groups translate directly into inequitable allocation of clinical resources, motivating the development of formal fairness criteria for healthcare machine learning.

Several competing fairness notions have been formalised in the literature, each encoding a distinct ethical commitment [6]. *Demographic parity* requires equal positive prediction rates across groups; *equalized odds* requires equal true positive and false positive rates; and *equal opportunity* requires equal true positive rates only [7]. Crucially, Verma and Rubin demonstrate that these definitions are mutually incompatible when base rates differ between groups [6].

Fairness interventions in the machine learning pipeline are categorised into three stages. Pre-processing methods modify the training data; instance reweighting, proposed by Kamiran and Calders, adjusts sample weights inversely proportional to class-group marginal products to reduce discrimination [8]. In-processing methods incorporate fairness constraints or regularisation terms directly into the training objective [9]. Post-processing methods adjust model outputs after training, the most common being threshold calibration, which shifts the classification decision boundary independently per group to equalise error rates [7].

Privacy in medical AI adds a further dimension of concern. Dwork and Roth formalise differential privacy (DP) as the canonical framework: a randomised mechanism $\mathcal{M}$ satisfies (ε, δ)-DP if its output distribution changes by at most a factor e^{ε} (plus an additive δ) when any single record is added to or removed from the dataset [10]. Mireshghallah et al. survey the application of DP and related privacy-preserving techniques to deep learning models and identify membership inference and model inversion as the principal attack vectors in medical settings [11].

2.2 TinyML

TinyML refers to the deployment of trained machine learning models on microcontrollers (MCUs) operating under strict resource budgets: typically 256 KB to 1 MB of flash, 64–512 KB of RAM, and inference latency targets in the single-digit millisecond range [2, 3]. The MLPerf Tiny benchmark has standardised evaluation protocols for this class of system, driving rapid progress in model compression and hardware-aware neural architecture design [3].

Model quantization is the primary compression technique that makes TinyML feasible. Post-training INT8 quantization maps 32-bit floating-point weights and activations to 8-bit integers using a calibration dataset, achieving up to a four-fold reduction in model size with typically less than 1% accuracy loss on moderately sized models [12]. However, whether these efficiency gains materialise at the sub-1,000-parameter scale characteristic of lightweight healthcare classifiers, where serialisation format overhead may exceed weight compression savings, has not been examined. TensorFlow Lite provides a full pipeline from Keras model training to MCU deployment via a FlatBuffer serialisation format with an optimised reference interpreter [13]. A persistent limitation of quantization research is that evaluation focuses almost exclusively on accuracy and inference speed, with no consideration of whether quantization alters group-level fairness properties.

2.3 Research Gap

Each of these bodies of literature addresses a different concern in isolation, without accounting for the joint effects of quantization, fairness intervention, and privacy-preserving preprocessing under embedded deployment constraints [3, 9, 10, 11].

No prior study has jointly evaluated multiple fairness metrics, privacy-inspired preprocessing strategies, and quantization methods on the same healthcare dataset under conditions representative of embedded deployment. Four specific interactions remain uncharacterised: whether INT8 quantization alters fairness disparities; whether the noise magnitudes required for privacy protection are compatible with stable fairness metrics; whether post-hoc threshold calibration remains the most effective fairness intervention when the model is constrained to embedded hardware; and whether the efficiency benefits of INT8 quantization scale predictably with model size, or whether a minimum parameter threshold is required for compression gains to outweigh serialisation overhead. This paper addresses all four questions through a controlled, multi-objective empirical study across two healthcare datasets of contrasting scale, bridging the gap between TinyML systems research and deployment-aware trustworthy AI evaluation in healthcare.

3 Methodology

This section describes the experimental framework used to investigate the research question.

3.1 Problem Formulation

The study is framed as a multi-objective empirical evaluation. Let $f_\theta : \mathbb{R}^d \to [0, 1]$ denote a classifier with parameters θ and input dimension d (8 for Pima, $\sim$2,400 for the hospital dataset), trained under noise level σ and evaluated at threshold τ. Four objectives are considered simultaneously:

$$\min_{\theta,\,\sigma,\,\tau} \big(-\mathrm{Acc}(\theta,\tau,\sigma),\ -\sigma,\ \Delta_{\mathrm{Fair}}(\theta,\tau,\sigma),\ \mathrm{Cost}_{\mathrm{Deploy}}(\theta,\sigma)\big), \tag{1}$$

where $-\mathrm{Acc}(\cdot)$ minimizes classification error, and $-\sigma$ maximizes the injected privacy noise. The remaining objectives are formulated as composite metrics:

- $\Delta_{\mathrm{Fair}}(\cdot) = \frac{1}{2}\big(\Delta_{\mathrm{DP}}(\cdot) + \Delta_{\mathrm{EO}}(\cdot)\big)$ represents the average demographic disharmony, combining both Demographic Parity (DP) and Equal Opportunity (EO) fairness gaps.
- $\mathrm{Cost}_{\mathrm{Deploy}}(\cdot) = \omega_1|\theta|_{\mathrm{KB}} + \omega_2\lambda(\cdot)$ represents the combined computational deployment cost, where $|\theta|_{\mathrm{KB}}$ is model size, $\lambda(\cdot)$ is latency, and ω_1, ω_2 are scaling factors used to normalize the differing units.

Because no single configuration minimises all four objectives simultaneously, the study characterises the empirical trade-off landscape and interprets which design choices most strongly influence which objectives.

3.2 Datasets and Preprocessing

Pima Indians Diabetes dataset. We use the Pima Indians Diabetes dataset [4] for all fairness, privacy, and threshold experiments. The dataset contains 768 samples drawn from a single-ethnic female population of Pima Indian heritage, with eight numerical features: number of pregnancies, plasma glucose concentration, diastolic blood pressure, triceps skin-fold thickness, two-hour serum insulin, body mass index (BMI), diabetes pedigree function, and age. The binary target variable indicates the presence of type 2 diabetes within five years of measurement. Several features encode physiologically impossible zero values representing missing measurements; zero values in the glucose, blood pressure, skin thickness, insulin, and BMI columns are replaced with `NaN` prior to imputation [4]. A `SimpleImputer` with median strategy is applied to each training fold independently, followed by a `StandardScaler` fitted exclusively on the training fold and applied to the test fold. The protected attribute is derived from the `Age` feature using a binary median split computed once on the full dataset before any cross-validation splitting, yielding a threshold of approximately 29 years. Samples at or below this threshold form Group 0 (younger, protected group; ≈50%) and samples above it form Group 1 (older; ≈50%). All eight features, including `Age`, are retained in the model input. Evaluation uses stratified 10-fold cross-validation (seed 42), giving approximately 691 training samples and 77 test samples per fold.

Diabetes 130-US Hospitals dataset. To characterise the scale-dependence of quantization efficiency, we additionally evaluate deployment on the Diabetes 130-US Hospitals dataset [5], which contains 101,766 hospitalisation records from 130 US hospitals over ten years. The binary target indicates early readmission within 30 days. The dataset comprises 47 raw features including a mixture of numeric variables (e.g. time in hospital, number of procedures) and categorical variables (e.g. discharge disposition, diagnosis codes); after one-hot encoding, the feature dimension is approximately 2,400. Numeric features are imputed with the column median and normalised with a `StandardScaler`; categorical features are imputed with the column mode and encoded with `OneHotEncoder`. All transformations are fitted on the training fold only. The protected attribute is a binary age split at 65 years, the threshold corresponding to Medicare eligibility in the United States: Group 0 is patients younger than 65 (≈57% of samples), Group 1 is patients aged 65 or older (≈43%). Evaluation uses stratified 5-fold cross-validation (seed 42) due to the larger dataset size; noise injection is *not* applied to this dataset (see Section 3.4).

Design Space. Table 1 summarises the full experimental design space. The two datasets contribute complementary axes: the Pima dataset is used for all fairness, privacy, and threshold analyses; the hospital dataset is used exclusively for quantization efficiency measurements and the fairness-under-quantization comparison at zero noise.

Table 1: Full experimental design space. The dataset column indicates which experiments each axis applies to (P = Pima, H = Hospital).

Axis	Levels	Data
Models	MLP, Logistic Regression (LR)	P
Model (quant.)	Larger MLP (∼155k params)	H
Noise σ	0.0, 0.05, 0.1, 0.2, 0.5	P
Quantization	FP32 Keras, FP32 TFLite, INT8 TFLite	P+H
Threshold τ	0.3, 0.4, 0.5 (default), 0.6, 0.7	P
Protected attr.	Age $\leq$ median / Age $>$ 65	P/H
Evaluation	10-fold CV (P), 5-fold CV (H), seed 42	P/H

3.3 Model Architectures

Model selection followed two criteria: interpretability and embedded suitability. Logistic regression was chosen as a transparent, well-understood baseline that isolates fairness and privacy effects from model complexity. The MLP was chosen as the primary model for its native compatibility with TFLite quantization (the central efficiency intervention) and its more consistent generalisation across folds (lower accuracy variance, Section 4.1), despite comparable discriminative power to the LR baseline. Alternative models, including XGBoost, were evaluated in preliminary experiments; XGBoost matched MLP accuracy but does not convert natively to TFLite, making it unsuitable for embedded deployment evaluation.

Pima MLP. The primary model for Pima experiments is a lightweight fully connected MLP. The architecture comprises an input layer of dimension 8, a first hidden layer of 12 neurons with ReLU activation, a dropout layer (rate 0.2), a second hidden layer of 8 neurons with ReLU activation, and a sigmoid output neuron. The trainable parameter count is 221: $8{\times}12{+}12 = 108$ (first layer), $12{\times}8{+}8 = 104$ (second layer), and $8{\times}1{+}1 = 9$ (output). The 12–8 configuration was selected after preliminary trials over hidden-layer widths $\{8, 12, 16\}$ and dropout rates $\{0.0, 0.2, 0.3\}$; it achieved accuracy comparable to the logistic regression baseline while keeping the total parameter count below 300, the practical upper bound for kilobyte-scale MCU flash budgets. All Pima models are trained for 100 epochs using the Adam optimiser (learning rate 10^{-3}) and binary cross-entropy loss, standard defaults for binary classification with sigmoid output, with convergence confirmed by monotonically decreasing training loss across folds. Figure 1 illustrates the architecture.

Hospital MLP. For the hospital dataset quantization experiments, a larger architecture is used to justify the higher-dimensional input (2,375 features after encoding) and to produce a model scale at which INT8 quantization efficiency can be assessed without format overhead dominating. The architecture is: Dense(64, ReLU) → BatchNormalisation → Dropout(0.3) → Dense(32, ReLU) → Dropout(0.3) → Dense(16, ReLU) → Sigmoid output. The trainable parameter count is approximately 154,817, dominated by the first layer ($2{,}375{\times}64{+}64 = 152{,}064$). The depth and width were chosen to accommodate the high-dimensional

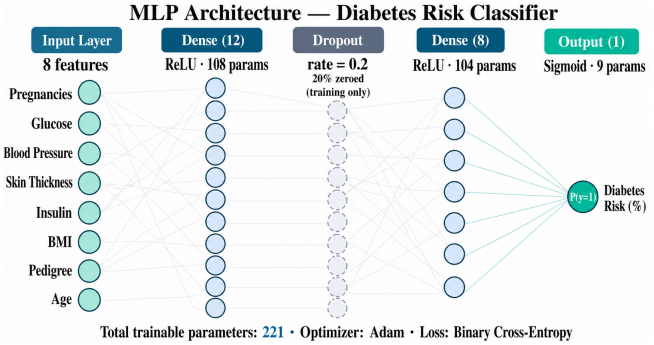

Figure 1: Pima MLP architecture for diabetes risk classification. Input: 8 clinical features. Hidden layers: Dense(12, ReLU) → Dropout(0.2) → Dense(8, ReLU). Output: sigmoid activation. Total: 221 trainable parameters.

one-hot-encoded input and to produce a parameter count large enough for INT8 compression gains to outweigh TFLite serialisation overhead; shallower alternatives (e.g. Dense(32) → Dense(16)) were found in preliminary experiments to fall below the scale at which INT8 efficiency materialises. Models are trained for 25 epochs with Adam (learning rate 10^{-3}), binary cross-entropy loss, and a batch size of 256, following standard practice for tabular classification at this scale; 25 epochs was sufficient for loss convergence given the larger dataset size. Figure 2 illustrates the architecture.

Logistic Regression (LR) baseline. The LR model [14] is used for Pima experiments only, serving as an interpretable baseline against which MLP generalisation and fairness behaviour can be compared. It uses ℓ_2 regularisation with $C = 3$, selected over the sklearn default ($C = 1$) after a brief search over $C \in \{0.1, 1, 3, 10\}$ based on balanced accuracy on held-out folds. The `liblinear` solver is the most efficient choice for ℓ_2-regularised binary classification on datasets with fewer than 10,000 samples [14]. Balanced class weighting corrects for the dataset's ≈35% positive-class imbalance, and the iteration limit is raised to 2,000 to ensure convergence.

3.4 Privacy Interventions

Noise injection. Gaussian noise injection adds independent perturbations drawn from $\mathcal{N}(0, \sigma^2)$ to each feature of every *training* sample, functioning as a practical approximation to the Gaussian mechanism of differential privacy [10]. Noise is applied after scaling and to the training set only, to avoid contaminating the test evaluation. Under the Gaussian mechanism, adding noise with standard deviation σ to a function of sensitivity Δf provides (ε, δ)-differential privacy, where:

$$\varepsilon = \frac{\Delta f \cdot \sqrt{2 \ln(1.25/\delta)}}{\sigma}. \tag{2}$$

For normalised features ($\Delta f = 1$, $\delta = 10^{-5}$), the five tested noise levels correspond to $\varepsilon \in \{\infty, 96.9, 48.5, 24.2, 9.7\}$. All levels remain in the weak-to-moderate regime ($\varepsilon \gg 1$) [10]. Noise injection is applied exclusively to the Pima dataset. For the hospital dataset, the ∼2,400-dimensional feature space means that even small σ values produce total perturbation power ($2{,}400 \times \sigma^2$) orders of magnitude larger than for Pima (8 features), making the noise levels non-comparable across datasets; hospital experiments therefore use $\sigma = 0$ throughout.

Gaussian noise injection was selected as the privacy proxy after comparing it against feature dropout (random zeroing of input features at training time); across the tested configurations, noise injection provided more stable accuracy–privacy–fairness trade-offs and maintains a direct, well-characterised connection to the formal Gaussian mechanism [10].

Threshold calibration. Post-hoc threshold calibration shifts the classification boundary by varying the decision threshold $\tau \in \{0.3, 0.4, 0.5, 0.6, 0.7\}$ applied to the model's sigmoid output. Threshold analysis is conducted on the Pima dataset only.

3.5 Embedded Deployment Setup

The setup deliberately separates training-time and inference-time environments to reflect realistic edge deployment conditions [2]. After training, each Keras model is exported as a TensorFlow Lite FlatBuffer using the `TFLiteConverter` API [13]. For FP32 TFLite configurations, the model is converted without quantization. For INT8 TFLite configurations, post-training quantization maps weights and activations to 8-bit integers using the min–max calibration scheme of Jacob et al. [12], calibrated on a representative subset of 200 training samples.

Post-training INT8 quantization was selected after preliminary experiments on the Pima model comparing quantization-aware training (QAT), a two-phase QAT variant, and INT4 post-training quantization. QAT variants produced no meaningful accuracy improvement at the Pima model scale relative to standard PTQ INT8, while INT4 introduced disproportionate accuracy degradation without additional efficiency gains under the TFLite reference interpreter. PTQ INT8 was therefore adopted as the standard compression technique, enabling straightforward evaluation of fairness and efficiency effects without confounding training-time modifications.

FP32 Keras model size and latency are measured on an x86-64 workstation running TensorFlow 2.x. FP32 and INT8 TFLite model size and latency are measured on a Raspberry Pi 4B using the TFLite reference interpreter. For the hospital model, latency benchmarks are conducted over 100,000 invocations per model with a 100-call warm-up phase. For the Pima model, fairness-under-quantization metrics (DP gap and EO gap) are evaluated on the TFLite interpreter for both FP32 and INT8 formats using the held-out test set. Because training and TFLite inference run on different hardware, all latency figures are reported as end-to-end deployment measurements.

Model size is the byte length of the serialised `.tflite` file (`os.path.getsize()`). Latency is the mean time per single-sample inference over the benchmark runs, using `time.perf_counter()`.

3.6 Evaluation Metrics

Evaluation metrics are organised into three groups.

Predictive performance. Accuracy, balanced accuracy, and AUC are computed on the held-out test fold.

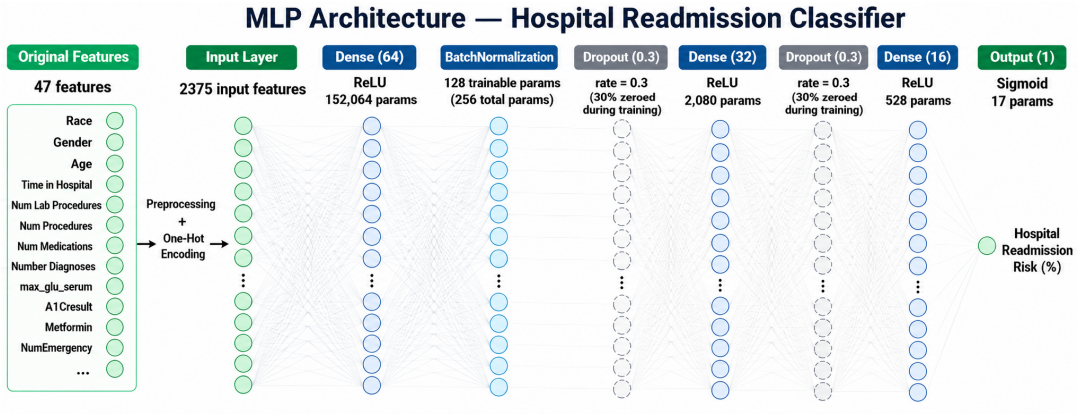

Figure 2: Hospital readmission MLP architecture for diabetes patient records. Initial: 47 clinical and demographic features. Hidden layers: Dense(64, ReLU) → BatchNormalization → Dropout(0.3) → Dense(32, ReLU) → Dropout(0.3) → Dense(16, ReLU). Output: sigmoid activation. Total: 154,817 trainable parameters.

Table 2: Baseline model performance and fairness across 10-fold CV at $\sigma = 0$ and $\tau = 0.5$. Both models show comparable discriminative power but non-trivial fairness disparities.

Model	Acc (%)	Bal. Acc	AUC	DP gap	EO gap
MLP	74.98 ± 4.95	0.757	0.835	0.310	0.177
LR	76.42 ± 5.96	0.754	0.837	0.308	0.152

Fairness. The *demographic parity gap* (DP gap) measures the absolute difference in positive prediction rates:

$$\Delta_{\mathrm{DP}} = \left|P(\hat{y}{=}1 \mid A{=}0) - P(\hat{y}{=}1 \mid A{=}1)\right|. \tag{3}$$

The *equal opportunity gap* (EO gap) measures the absolute difference in true positive rates [7]:

$$\Delta_{\mathrm{EO}} = \left|\mathrm{TPR}_0 - \mathrm{TPR}_1\right|, \tag{4}$$

where $\mathrm{TPR}_g = P(\hat{y}{=}1 \mid y{=}1, A{=}g)$. Lower values indicate greater fairness. The EO gap is the primary criterion, as equal TPR rates are clinically more meaningful than equal prediction rates in a screening context.

Embedded efficiency. Model size is reported in kilobytes (KB) and inference latency in milliseconds (ms).

4 Results

This section presents the experimental results in six stages.

4.1 Baseline Performance and Fairness

Table 2 reports 10-fold cross-validation performance at zero noise ($\sigma = 0$) and the default threshold ($\tau = 0.5$). The MLP and LR baselines achieve comparable AUC, indicating similar discriminative power. However, the MLP generalises more consistently across folds, evidenced by lower accuracy variance, likely due to dropout regularisation. Both models exhibit non-trivial fairness disparities at baseline, which serve as the reference point for all subsequent interventions.

4.2 Privacy Effects on Accuracy and Fairness

The effect of Gaussian noise injection on the LR model is detailed in Table 3 and Figure 3. Accuracy decreases monotonically with noise, with the sharpest decline occurring between $\sigma = 0.05$ and $\sigma = 0.10$.

In contrast, fairness metrics exhibit non-monotonic behaviour. As shown in Table 3, the EO gap fluctuates significantly across noise levels. This instability indicates that noise injection does not improve group fairness predictably; instead, perturbations affect age groups at different rates, creating an unstable privacy–fairness relationship. As computed in Section 3.4, the five noise levels correspond to $\varepsilon \in \{\infty, 96.9, 48.5, 24.2, 9.7\}$ under the Gaussian mechanism, all remaining in the weak-to-moderate privacy regime ($\varepsilon \gg 1$) [10]. The $\sigma = 0.05$ level provides the most favourable overall trade-off: negligible accuracy loss (<0.01 pp), the lowest EO gap of any tested configuration (0.141), and a non-trivial privacy proxy ($\varepsilon \approx 96.9$).

The non-monotonic fairness behaviour stems from asymmetric feature distributions between age groups: younger patients (Group 0) exhibit lower glucose and BMI values, positioning them differently in feature space. At low magnitudes ($\sigma = 0.05$), noise acts as a mild regulariser, smoothing group-correlated feature reliance and marginally equalising true positive rates. At $\sigma = 0.10$, perturbations are large enough to destabilise the marginal positive predictions of Group 0, significantly widening the EO gap, empirical evidence for the theoretical privacy–fairness conflict identified by Dwork and Roth [10].

4.3 Quantization Effects on Fairness and Efficiency

This section reports quantization results for both datasets. Embedded efficiency is evaluated on the hospital readmission model (∼154,800 parameters), where the model scale is sufficient to demonstrate genuine compression gains. Fairness effects are reported for both models, revealing scale-dependent behaviour.

Embedded efficiency. Table 4 compares FP32 TFLite and

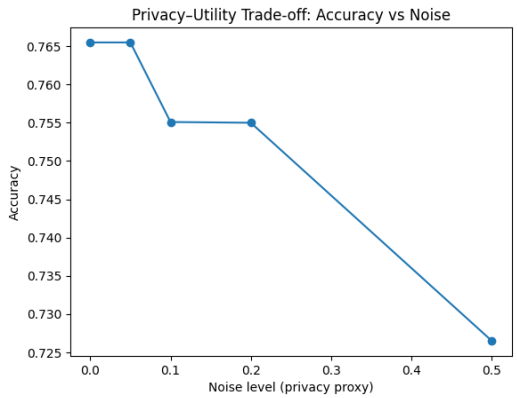

Figure 3: Privacy–utility trade-off: accuracy decreases monotonically with noise level, with the sharpest drop between $\sigma = 0.05$ and $\sigma = 0.10$.

Table 3: Effect of Gaussian noise injection on accuracy and fairness (LR, Pima dataset, 10-fold CV). The EO gap fluctuates non-monotonically, confirming a privacy–fairness tension rather than a linear trade-off.

Noise σ	Accuracy	DP gap	EO gap
0.00	0.7655	0.3087	0.1528
0.05	0.7654	0.3031	0.1413
0.10	0.7551	0.3036	0.1737
0.20	0.7550	0.3267	0.1488
0.50	0.7265	0.3030	0.1667

Table 4: Scale-dependent quantization efficiency: FP32 TFLite vs INT8 TFLite for both dataset models. Pi latency measured over 100,000 invocations. INT8/FP32 ratio < 1 indicates INT8 is smaller; Speedup < 1 indicates INT8 is slower.

Dataset	Params	FP32 (KB)	INT8 (KB)	INT8/FP32 ratio	FP32 Pi (ms)	INT8 Pi (ms)	Speedup
Pima	221	2.93	3.10	1.06	0.012	0.079	0.15×
Hospital	154,817	613.05	158.50	0.26	0.091	0.025	3.64×

INT8 TFLite across both dataset scales. For the Pima model (221 parameters), INT8 TFLite (3.10 KB) is marginally *larger* than FP32 TFLite (2.93 KB), and inference on the Raspberry Pi 4 is approximately $6.5\times$ *slower* (0.079 ms vs 0.012 ms). Both effects are attributable to the FlatBuffer quantization metadata overhead exceeding the weight compression saving at this parameter count, and to the absence of hardware INT8 SIMD acceleration for sub-kilobyte models on the Pi reference interpreter. For the hospital model (154,817 parameters), the outcome reverses entirely: INT8 TFLite (158.50 KB) is $3.87\times$ *smaller* than FP32 TFLite (613.05 KB), and Pi inference is $3.6\times$ *faster* (0.025 ms vs 0.091 ms), consistent with the four-fold theoretical compression of 32-bit to 8-bit weights. These results establish that INT8 quantization efficiency is scale-dependent: a minimum parameter threshold must be exceeded for weight savings to outweigh serialisation overhead.

Fairness under quantization. Table 5 reports MLP accuracy and fairness for the Pima model under FP32 and INT8 TFLite across noise levels. At $\sigma = 0$, INT8 incidentally reduces the EO gap (0.177 to 0.162), but at all non-zero noise levels INT8 consistently amplifies disparities: at $\sigma = 0.5$, the INT8 EO gap (0.369) is 66.8% higher than the FP32 EO gap (0.221). This confirms that INT8 quantization exacerbates the privacy–fairness tension at the Pima model scale.

Table 5: Pima MLP fairness and accuracy under FP32 vs. INT8 quantization across noise levels. INT8 amplifies both DP and EO gaps at all non-zero noise levels, with the most severe impact at $\sigma = 0.5$.

σ	FP32 Acc	FP32 DP	FP32 EO	INT8 Acc	INT8 DP	INT8 EO
0.0	0.754	0.310	0.177	0.732	0.324	0.162
0.05	0.754	0.320	0.180	0.741	0.327	0.180
0.1	0.754	0.323	0.192	0.745	0.349	0.245
0.2	0.759	0.329	0.220	0.732	0.351	0.279
0.5	0.749	0.244	0.221	0.719	0.352	0.369

Table 6: Threshold sensitivity analysis (Pima dataset, averaged across models and noise levels). $\tau = 0.7$ achieves the lowest disparity at minimal accuracy cost.

Threshold τ	Accuracy	DP gap	EO gap
0.3	0.747	0.339	0.138
0.4	0.763	0.297	0.143
0.5	0.764	0.253	0.152
0.6	0.771	0.233	0.208
0.7	0.753	0.167	0.124

This asymmetric amplification occurs because younger patients (Group 0) have a lower base rate of diabetes, placing more of their positive predictions near the decision boundary; INT8 weight rounding perturbs this boundary non-uniformly, disproportionately destabilising Group 0 predictions when combined with noise injection.

For the hospital model at $\sigma = 0$, the direction reverses: FP32 TFLite achieves accuracy 0.882, DP gap 0.008, and EO gap 0.041 (identical to the Keras model), while INT8 TFLite achieves the same accuracy (0.882) with DP gap 0.004 and EO gap 0.016, a 61% reduction in equal opportunity gap at zero accuracy cost. This improvement is attributable to the larger, heavily regularised model having a smoother decision boundary: weight rounding at this scale homogenises predictions across groups rather than distorting them asymmetrically. Noise injection is not applied to the hospital model (see Section 3.4), so the fairness-under-quantization interaction at non-zero noise cannot be assessed for this dataset.

4.4 Threshold Sensitivity Analysis

Table 6 and Figure 4 demonstrate the impact of classification thresholds (Pima dataset, averaged across models and noise levels). A non-monotonic trend is observed for the EO gap: raising τ to 0.6 improves accuracy and DP but significantly worsens EO. This occurs because the reduction in positive predictions disproportionately affects the younger protected group. However, at $\tau = 0.7$, both fairness metrics reach their minimum values with minimal accuracy cost. This post-hoc calibration provides the dominant fairness gain of the study without incurring embedded overhead.

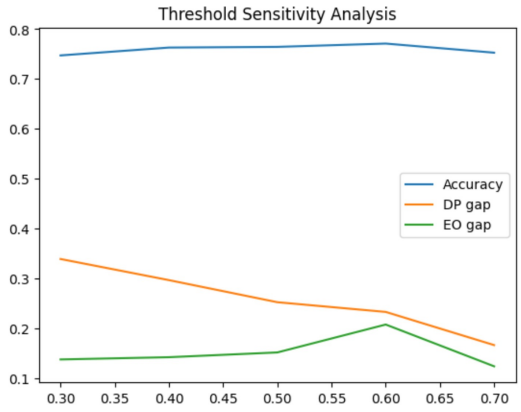

Figure 4: Threshold sensitivity: DP gap decreases monotonically, while EO gap peaks at $\tau = 0.6$ before reaching its minimum at $\tau = 0.7$.

4.5 Multi-Objective Trade-off Landscape

Pareto analysis (Figure 5) using LR identifies three Pareto-optimal noise configurations on the Pima dataset: $\sigma \in \{0.0, 0.05, 0.5\}$. While $\sigma = 0.0$ and $\sigma = 0.5$ represent non-dominated theoretical extremes (highest accuracy and marginally lowest DP gap respectively), $\sigma = 0.05$ offers the best practical fairness profile: the lowest EO gap (0.141) at effectively unchanged accuracy. Intermediate levels $\sigma = 0.10$ and $\sigma = 0.20$ are dominated, incurring accuracy loss without consistent fairness improvement.

Taken together, the results indicate that combining low-magnitude noise ($\sigma = 0.05$) with threshold calibration ($\tau = 0.7$) is the most effective strategy for the Pima embedded healthcare setting. INT8 quantization provides no efficiency benefit at the Pima model scale and actively exacerbates fairness disparities under noise; at the larger hospital scale (Section 4.3), efficiency gains are substantial but the noise–fairness interaction cannot be characterised due to feature dimensionality constraints.

4.6 Interactive Trade-off Analysis

To navigate the complex trade-offs between predictive performance, algorithmic bias, and resource constraints, an interactive model selection dashboard was developed (Figure 6). This tool allows stakeholders to assign custom weights to accuracy, fairness, privacy, and deployment efficiency based on specific clinical priorities. The system dynamically ranks all candidate models across FP32 and INT8 formats and calculates a weighted composite score. By visualizing these multi-objective trade-offs, the tool operationalizes the theoretical findings of this research, enabling informed, context-specific model deployment rather than relying on a single static performance metric.

5 Responsible Research

The Pima dataset is drawn from a single-ethnic female population of Pima Indian heritage aged 21 or older; models trained on it must not be extrapolated to other demographics without recalibration. The Hospital readmission dataset

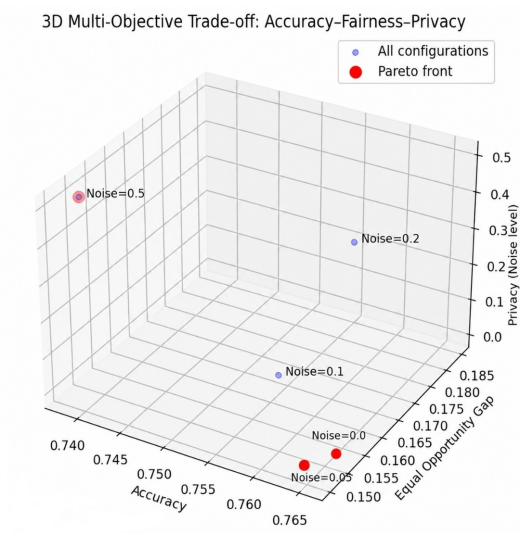

Figure 5: Multi-objective trade-off space (Pima dataset) using LR. Three configurations are Pareto-optimal; $\sigma = 0.05$ provides the best combined fairness profile with near-maximal accuracy.

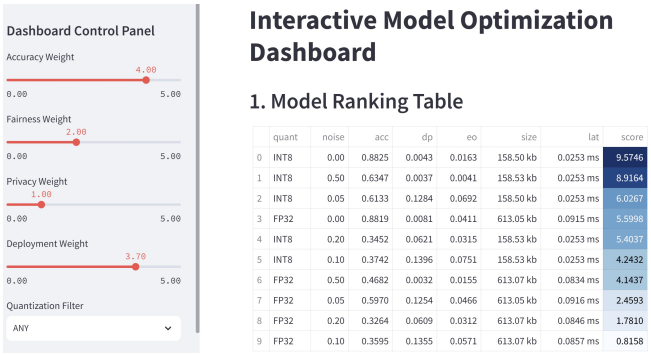

Figure 6: Interactive dashboard prototype allowing stakeholders to assign weights to accuracy, fairness, privacy, and deployment metrics to dynamically rank candidate models.

carries a direct resource-allocation risk of its own: misclassifying a patient as low-risk for early readmission could lead to premature discharge and adverse outcomes, so the legally grounded age-65 threshold alone does not justify clinical deployment without further review. For both datasets, age is used as the protected attribute as a methodological simplification, legally protected attributes in healthcare typically include race, sex, and disability status, and their selection requires domain-expert consultation with affected communities. The finding that INT8 quantization can amplify fairness disparities by up to 66.8% underscores that fairness auditing must be a mandatory step in any embedded deployment pipeline, not an optional post-hoc consideration. All code, notebooks, and results are openly available[1]. Experiments are fully reproducible under fixed seed 42 using publicly avail-

[1]The full source code, along with instructions, is available at: https://github.com/tompealuca/healthcare_trustworthy_embedded_AI

able datasets [4, 5], with the caveat that Pi 4B latency figures are representative hardware measurements rather than deterministic constants.

AI tools were used in a supporting role only: debugging experimental Python code, improving grammatical clarity, and resolving LaTeX typesetting issues. Appendix A provides a representative selection of prompts submitted to AI tools during the project, illustrating the nature and scope of their use. All scientific decisions — research question, experimental design, dataset and metric selection, result interpretation, and conclusions — were made independently by the author.

6 Discussion

Scale-dependent priorities. The results establish that embedded deployment constraints do not introduce new fairness–accuracy trade-offs but clarify the design priority space in a scale-dependent way. At sub-1,000-parameter MCU scale, computationally expensive in-processing regularisers are infeasible, making threshold calibration the dominant zero-cost intervention; at larger TinyML scale, INT8 quantization can improve both efficiency and fairness simultaneously. In both cases, the direction of the quantization fairness effect cannot be predicted without empirical evaluation, making post-deployment fairness auditing mandatory [3, 9].

Dataset generalisability. The Pima dataset is small (768 samples), single-ethnic, and restricted to female participants over 21. Although the hospital readmission dataset addresses scale, it represents a different clinical task and feature space; direct generalisation of the fairness and privacy findings across both datasets is therefore limited.

Privacy proxy limitations. Gaussian noise injection is a heuristic privacy proxy, not a certified (ε, δ)-DP mechanism [10]. The ε values are computed under the assumption $\Delta f = 1$, and even $\varepsilon \approx 9.7$ provides weak formal protection against adversaries with auxiliary data. Furthermore, applying equal σ to the 2,400-dimensional hospital feature space would produce noise power orders of magnitude larger than for Pima, making direct cross-dataset noise comparisons methodologically invalid.

Hardware measurement caveats. Hardware evaluation relies on a Raspberry Pi 4B running a general-purpose OS, which introduces scheduling variability; reported latency figures are representative measurements rather than deterministic constants. A fully reproducible evaluation would require bare-metal MCU deployment with hardware-interrupt-based timing and energy profiling.

Further limitations. Finally, the absence of a held-out validation set limits the strength of absolute performance claims, though comparative findings remain valid since all configurations are evaluated under identical conditions. Results are restricted to age-stratified binary classification on tabular health data; generalisation to other protected attributes, modalities, or clinical tasks requires further investigation.

7 Conclusion and Future Work

This study showed that INT8 quantization and privacy-preserving noise affect fairness and efficiency in healthcare TinyML models in scale-dependent ways that cannot be assumed without empirical evaluation. At sub-300 parameters, INT8 provides no efficiency benefit and amplifies fairness disparities by up to 66.8% under noise; at $\sim$155k parameters, it achieves 3.87$\times$ compression, 3.6$\times$ speedup, and a 61% EO gap reduction. Low-magnitude noise (σ=0.05) is a safe privacy proxy, while higher levels create an unpredictable non-monotonic privacy–fairness tension. Post-hoc threshold calibration (τ=0.7) is the dominant zero-overhead fairness lever, yielding an 18.4% relative EO gap reduction at 1.2 pp accuracy cost [7]. Within the conditions examined, embedded constraints do not introduce new fairness–accuracy trade-offs but shift design priorities toward post-deployment calibration and mandatory fairness auditing.

Future work should validate these findings across diverse clinical datasets and protected attributes, replace heuristic noise proxies with formal DP-SGD [15], and extend hardware evaluation to bare-metal ARM Cortex-M with energy-aware benchmarking.

References

[1] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453, 2019.

[2] Pete Warden and Daniel Situnayake. *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. O'Reilly Media, 2019.

[3] Colby Banbury et al. MLPerf tiny benchmark. In *Proceedings of the NeurIPS Datasets and Benchmarks Track*, 2021.

[4] Jack W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler, and R. S. Johannes. Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, pages 261–265, 1988.

[5] Beata Strack, Jonathan P. DeShazo, Chris Gennings, Juan L. Olmo, Sebastian Ventura, Kristen J. Pham, and John N. Clore. Impact of HbA1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records. *BioMed Research International*, 2014:781670, 2014.

[6] Sahil Verma and Julia Rubin. Fairness definitions explained. In *Proceedings of the IEEE/ACM International Workshop on Software Fairness (FairWare)*, pages 1–7, 2018.

[7] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, 2016.

[8] Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1):1–33, 2012.

[9] Rachel K. E. Bellamy et al. AI Fairness 360: An extensible toolkit for detecting and mitigating algorith-

mic bias. *IBM Journal of Research and Development*, 63(4/5):4:1–4:15, 2019.

[10] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[11] Fatemehsadat Mireshghallah et al. Privacy in Deep Learning: A Survey. *arXiv preprint arXiv:2004.12254*, 2020.

[12] Benoit Jacob et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2704–2713, 2018.

[13] TensorFlow Authors. TensorFlow Lite: Machine learning for mobile and edge devices. https://www.tensorflow.org/lite, 2023.

[14] Fabian Pedregosa et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[15] Martin Abadi et al. Deep learning with differential privacy. In *ACM Conference on Computer and Communications Security (CCS)*, 2016.

A Representative AI Tool Prompts

This appendix provides a representative selection of prompts submitted to AI tools during the development of this project. The examples are organised by use category and illustrate the scope and nature of AI assistance described in Section 5. In all cases, the author verified the outputs independently; no AI-generated content was incorporated without review.

A.1 Code Debugging

The following prompts were used to identify implementation errors after the author had already diagnosed the likely source of the problem:

1. *"In my cross-validation loop, I am computing the split attribute as* `(X_test[:, 0] > np.median(X_test[:, 0]))` *but I get different fairness values each fold. Why does using the median cause this, and how do I fix it using a global median computed before splitting?"*
2. *"My* `add_noise` *function adds Gaussian noise to both* `X_train` *and* `X_test`*. I think this is wrong for a privacy proxy. Can you confirm, and show me the correct way?"*
3. *"The fairness values for FP32 and INT8 in my quantization experiment are identical. I think the INT8 predictions are not being used, can you check this code and identify where the predictions go wrong?"*

A.2 Writing Clarity

The following prompts were used to improve the readability of written sections after the author had drafted the initial content:

1. *"Here is my paragraph describing the sensitive attribute definition. Can you improve the clarity without changing the technical content or adding new claims?"*
2. *"Is this paragraph written in a formal academic tone for a computer science research paper? If not, which phrases should I revise?"*

A.3 LaTeX Formatting

The following prompts were used to resolve typesetting issues:

1. *"My table is overflowing the column width in a two-column IEEE-style layout. Here is the latex code, what is the cause and how do I fix it?"*
2. *"How do I write the Gaussian mechanism equation for differential privacy in latex, with the special symbols variables?"*
3. *"Can you generate an example BibTeX entry for my paper so I see the structure?"*