International Journal of Future Engineering Innovations

Assessing the Transferability of Deep Learning from Synthetic to Real Clinical Data for Heart Disease Prediction: An Architecture-Controlled Comparative Study

Yasser Hassen Al-Maamori
Babylon Polytechnic College, Al-Furat Al-Awsat Technical University, Babil 51015, Iraq

* Corresponding Author: **Yasser Hassen Al-Maamori**

Article Info

ISSN (online): 3049-1215
Volume: 03
Issue: 06
Received: 22-08-2026
Accepted: 24-09-2026
Published: 26-10-2026
Page No: 31-43

Abstract
Synthetic patient records are increasingly used where real clinical data are scarce, restricted by privacy rules, or expensive to label. How far performance on such data predicts performance on genuine records remains unclear for tabular cardiac prediction. This study fixes the architecture and varies only the data. One feed-forward network, unchanged in structure, was trained on a synthetic corpus of 9,000 patient records and on the real UCI Cleveland Heart Disease dataset of 303 patients, under identical preprocessing, class-weighted optimization, and evaluation, so that any difference reflects data provenance rather than model design. On the synthetic data the network reached an accuracy of 0.869 and an AUC of 0.940. Transferred to the real data, accuracy dropped to 0.787 while AUC stayed at 0.920: the model still ranked patients well, but its calibrated classification deteriorated. A second experiment on the real data added early stopping, five-fold stratified cross-validation, a hyperparameter grid search and a fold-level ensemble. Accuracy returned to 0.869 and AUC rose to 0.953, in line with published results on the same benchmark. The search selected the most strongly regularized configuration, which points to overfitting on a small sample rather than any intrinsic difficulty of the real data. Two practical implications follow. Synthetic data can make a classification task look easier than it is, and the gap this creates is largely closed by standard regularization. Synthetic datasets are suitable for model prototyping; however, conclusions derived from them must be validated using real clinical data before clinical interpretation.

DOI: https://doi.org/10.54660/IJFEI.2026.3.6.31-43

Keywords: heart disease prediction, deep learning, synthetic clinical data, UCI Cleveland, multilayer perceptron, cross-validation

1. Introduction

Cardiovascular disease remains one of the principal contributors to global mortality and disability, placing a sustained demand on preventive and curative healthcare services [1]. Identifying individuals at elevated risk before the onset of acute events allows earlier intervention and a more efficient distribution of limited clinical resources. This clinical need has driven decades of work on computational risk prediction, progressing from regression-based scores with a small number of predefined variables toward machine learning and, more recently, deep learning models capable of representing intricate, non-linear dependencies among many predictors [2,3,4,5].

The behaviour of any data-driven cardiac model is bounded by the data available for its development. Real clinical datasets are frequently small, imbalanced, incompletely labelled, and difficult to share because of privacy regulation and institutional governance. These constraints have stimulated interest in synthetic data: artificially generated records that emulate the statistical structure of authentic cohorts while reducing direct exposure of patient information. Synthetic datasets are now widely

distributed for teaching, prototyping, and benchmarking, and generation techniques have advanced from simple marginal sampling to generative adversarial networks [6], variational autoencoders [7], and conditional models for tabular data [8].

This growing reliance introduces a methodological question that has received limited empirical treatment in the cardiac tabular setting. Strong performance on a synthetic dataset does not guarantee that the same model, or the same reported metrics, will hold on real patients. A generator may produce dependencies that are more regular and therefore easier to classify than those observed clinically, yielding optimistic estimates; alternatively, it may introduce artefacts that depress performance. In the absence of a controlled comparison, the transferability of synthetic-data conclusions to real data remains uncertain, and reports based only on synthetic data may misrepresent clinical readiness.

The gap in the existing literature is specific. A large body of work evaluates the statistical fidelity of synthetic data by comparing marginal and joint distributions against real cohorts, and a separate body of work reports predictive performance of deep models on real cardiac datasets. Comparatively little work joins these threads by holding a predictive model fixed and measuring how its performance changes when only the data provenance is varied. Studies that do compare synthetic and real data often change the model, the feature set, and the evaluation protocol simultaneously, which entangles the effect of data provenance with modelling choices. An experiment that fixes the architecture and the protocol, varying only the data, is therefore needed to attribute performance differences to provenance with confidence. Supplying such an experiment for cardiac tabular prediction is the primary aim of this study.

The contributions of this study are fourfold. First, it provides an architecture-controlled comparison of synthetic and real clinical data for heart disease prediction, removing model choice as a confounding factor. Second, it quantifies the transfer gap using complementary metrics and shows that discrimination and calibrated accuracy respond differently to data provenance. Third, it demonstrates through a controlled optimization study that the gap is driven by overfitting on the small real sample and can be closed with standard techniques. Fourth, it situates the results within the published literature on the UCI Cleveland benchmark, confirming external consistency and interpreting the synthetic result without conflating it with real-data performance.

2. Literature Review

2.1. Machine and Deep Learning for Cardiac Prediction

The UCI Cleveland cohort has served as a reference benchmark for cardiac classification for more than three decades, and the accumulated results define a useful expectation range. A hybrid model combining a random forest with a linear learner reported an accuracy near 88.7%, arguing that hybridizing complementary learners captures both axis-aligned and linear structure [9].

An evaluation of K-nearest neighbours, logistic regression, and random forests obtained a mean accuracy around 87.5% and noted sensitivity to feature selection [10]. A K-nearest-neighbour classifier reached 90.8% [11], while random forests with feature selection achieved 88.5% [12]. Combined machine-learning and deep-learning pipelines have likewise been reported for the same benchmark [13]. In [14] evaluated several machine learning and deep learning approaches for coronary heart disease prediction and reported an accuracy of 92.42% using a support vector machine (SVM) classifier on the UCI Heart Disease dataset.

Deep learning approaches have expanded the design space. Comprehensive reviews covering 2018 to 2023 record a shift from single feed-forward networks toward integrated architectures that couple deep models with feature selection, optimization, or attention [15]. One study augmented cardiac features with a sparse autoencoder before classification, improving precision on limited data, an approach conceptually aligned with the regularization emphasis of the present work [16]. Another combined ensemble deep learning with feature fusion across the Cleveland and Hungarian cohorts, reporting high accuracy but relying on refined data handling that complicates comparison [17]. Recent work has also assembled modular pipelines that select among multilayer perceptrons, one-dimensional convolutional networks, and tabular transformers according to dataset characteristics, reporting an area under the curve near 0.90 on the same 303-record Cleveland benchmark under nested cross-validation [18]. Across these studies, reported accuracies concentrate in the high 80s to mid-90s, with the upper values frequently associated with aggressive feature engineering or with validation protocols that do not fully isolate model selection from testing.

The relevance of these methods extends beyond cardiac data to clinical tabular prediction more broadly. Machine learning with systematic hyperparameter optimization has been applied to liver disease prediction, where tuning and ensemble tree models materially improved predictive accuracy on tabular medical records [19]. That finding parallels the present study, in which hyperparameter search and ensembling proved decisive for performance on the small real dataset. In the imaging domain, deep convolutional networks have been applied to diagnostic tasks such as brain tumour detection from magnetic resonance images [20], illustrating the breadth of deep learning in healthcare while underscoring that tabular and image modality pose distinct modelling challenges. The present work is situated within the tabular branch, where sample size and regularization, rather than spatial feature extraction, dominate performance.

A consistent methodological caution in this literature is that small datasets are prone to optimistic evaluation. When hyperparameters are tuned on the same data used for final reporting, or when no untouched hold-out set is maintained, accuracy estimates can be inflated. Several authors therefore recommend cross-validation with explicit separation of tuning and testing. The protocol adopted here follows that recommendation and treats the reported figures as estimates obtained under a strict procedure rather than as best-case values.

2.2. Synthetic Clinical Data and Its Validation

Synthetic data generation has matured into a distinct research area within healthcare informatics. Several reviews trace the progression from marginal-based methods to generative adversarial networks, variational autoencoders, and diffusion models, and they stress that the practical value of synthetic data depends on downstream utility rather than statistical resemblance alone [21 ,22 ,23]. Systematic reviews of tabular health-record synthesis reach similar conclusions regarding the primacy of downstream utility [24].

The utility question has been formalized as whether models trained on synthetic data perform comparably to those trained

on real data, and whether comparisons made on synthetic data remain valid on real data.

Empirical evidence is mixed and domain dependent. In medical imaging, training partly or wholly on synthetic samples can improve robustness, whereas in tabular clinical settings the transfer is less predictable. A comparison of analyses on real and synthetic derivatives of the same cohort found broad but imperfect agreement, which motivates task-specific validation [25]. On tabular cardiac data specifically, a recent study used Bayesian-network generators to synthesize records from the UCI Cleveland cohort and evaluated real-only, synthetic-only, and hybrid training configurations, finding that the benefit of synthetic augmentation depended strongly on the structure and completeness of the source data and was largest for the noisier Switzerland cohort [26].

At larger scale, a dual adversarial autoencoder trained on nearly one million diabetes patients produced synthetic data on which models matched real-data predictive performance, yet sex-stratified analyses revealed biases invisible to standard utility metrics, reinforcing that aggregate parity can mask clinically relevant discrepancies [27]. A transfer-oriented study on colorectal cancer survival reached a complementary conclusion: a model pretrained on synthetic data and adapted with local hospital data matched a hospital-only baseline, but the authors emphasized that synthetic data complement rather than replace real clinical data [28]. Despite this progress, comparatively few studies place a fixed predictive model on both synthetic and real clinical data under identical conditions and analyze the resulting difference. The present study contributes such a controlled comparison for cardiac tabular prediction and links it to the practical problem of overfitting on small real datasets. Table 1 summarizes representative studies and positions the present work relative to them.

Table 1: Representative studies and the position of the present work.

Study	Data	Method	Focus / Limitation
Mohan *et al.*	UCI Cleveland	Hybrid RF + linear	High accuracy; single split
Shah *et al.*	UCI Cleveland	KNN	Strong result; small sample
Demir *et al.*	UCI Cleveland	SVM	Classification performance (92.42% accuracy)
Lazzaro *et al.*	UCI (3 cohorts)	Bayesian-network synthesis	Synthetic augmentation depends on data quality
Ortuño *et al.*	1M diabetes cohort	Dual adversarial auto encoder	Parity hides sex-stratified bias
Jang *et al.*	Synthetic + hospital	Domain adaptation / ensemble	Synthetic complements, not replaces
This work	Synthetic + UCI	Architecture-controlled MLP	Synthetic-to-real transfer

3. Theoretical Background

This section establishes the conceptual and mathematical basis for the model before its application is described. The prediction task is framed as supervised binary classification. Given a set of labelled examples, each consisting of a feature vector and a binary outcome indicating the absence or presence of heart disease, the objective is to learn a function that maps feature vectors to outcome probabilities and generalizes to unseen patients.

3.1. Supervised Classification Framework

Let each patient be represented by a feature vector and an associated label drawn from an unknown joint distribution over features and outcomes. A classifier is a parameterized function that estimates the conditional probability of disease given the features. Learning consists of selecting parameters that minimize a measure of discrepancy between predicted probabilities and observed labels on a finite training sample, subject to the expectation that the same parameters will perform well on independent data from the same distribution. The central difficulty is generalization: a model with sufficient capacity can fit the training sample closely while failing on new cases, a phenomenon known as overfitting that becomes more pronounced as sample size decreases relative to model capacity.

3.2. Feed-Forward Neural Networks as Universal Approximators

A feed-forward neural network composes alternating linear transformations and non-linear activation functions. With at least one hidden layer and a non-linear activation, such networks are universal approximators, able to represent any continuous function on a compact domain to arbitrary accuracy given sufficient width [29]. This expressive capacity motivates their use for clinical prediction, where relationships among risk factors are non-linear and interacting. The same capacity, however, makes explicit regularization essential when the training sample is small, because an unconstrained network can memorize idiosyncratic patterns that do not generalize. The methods developed later in the paper are chosen specifically to control this capacity.

3.3. The Bias–variance Perspective on Small Clinical Datasets

The behavior observed in this study is well described by the bias–variance decomposition of generalization error [30]. A flexible model applied to a large, regular dataset can achieve both low bias and low variance. Applied to a small dataset, the same model tends to exhibit high variance: its predictions depend strongly on the particular training sample, and its decision boundary, although correctly oriented, may be poorly located. Regularization techniques such as dropout, weight decay, early stopping, and ensembling reduce variance at the cost of a modest increase in bias, improving expected performance on unseen data. This perspective frames the empirical contrast between the large synthetic dataset and the small real dataset and explains why the optimization protocol closes the observed gap.

4. Experimental Setup

4.1. Dataset Preparation

Two datasets of contrasting provenance and scale were used, summarized in Table 2. The synthetic dataset is a publicly available Kaggle corpus of 9,000 generated records, described by its authors as synthetic with an approximate disease prevalence of 30%. Each record contains 25 predictive attributes spanning demographics, resting blood

pressure, a lipid and metabolic panel, cardiac indicators, and lifestyle factors, together with a binary label. It contains no missing values or duplicates, a characteristic of generated data. The real dataset is the UCI Cleveland Heart Disease dataset [31 ,32] of 303 patients with 13 clinical attributes; its original 0–4 severity label was binarized into absence versus presence of disease, and a small number of missing values were imputed.

Table 2: Summary of the two datasets.

Characteristic	Synthetic	UCI Cleveland
Records	9,000	303
Predictive features	25	13
Missing values	None	Few (imputed)
Disease prevalence	~30%	~46%
Provenance	Generated	Real clinical
Target encoding	Binary	Binarized (0 vs 1–4)

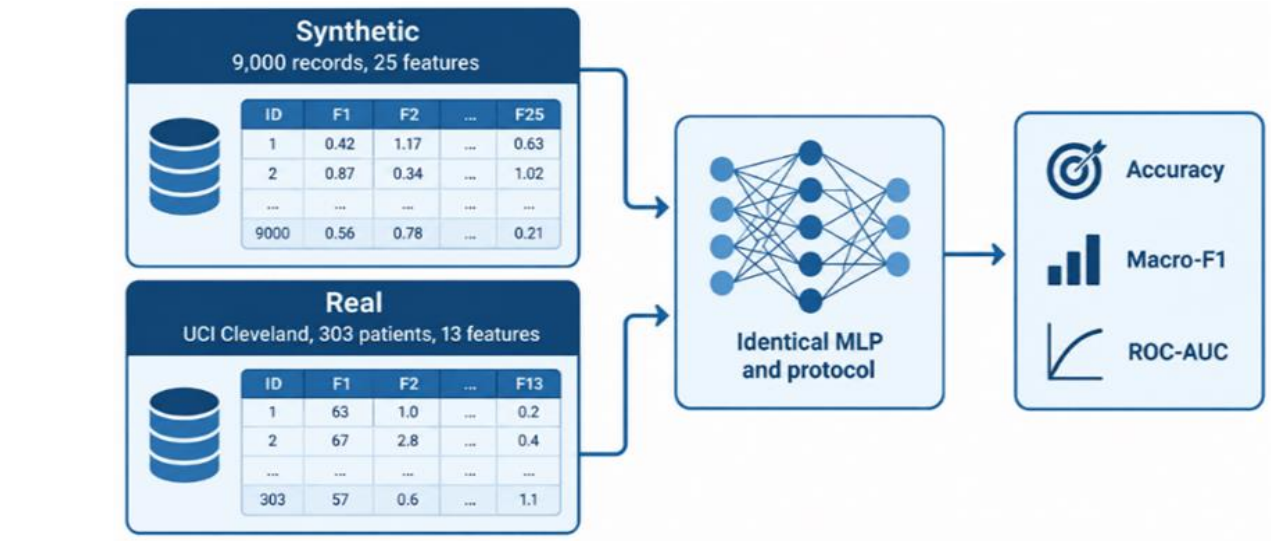


Fig 1: Overview of the two datasets feeding a single shared model.

4.2. Preprocessing

Preprocessing followed identical logic for both datasets. Non-numeric attributes were label-encoded. For the real dataset, missing entries were imputed using the median of the corresponding feature, a robust choice for skewed clinical variables. Each dataset was partitioned into training and test subsets by stratified sampling, reserving 20% for testing, and features were standardized using statistics computed on the training partition only. The formal definition of standardization is given in Section 5.

The choice of median imputation rather than mean imputation reflects the distributional characteristics of the affected variables. The vessel-count and thalassemia fields in the real dataset are discrete and asymmetric, so the median preserves a plausible central value without being distorted by extreme observations. Because only a small fraction of records required imputation, the procedure had limited influence on the overall feature distributions and computing the imputation values on the training partition alone avoided leakage of test information. Label encoding was preferred over one-hot encoding for the categorical fields because the network's first linear layer, followed by standardization, can absorb ordinal encodings without a large increase in input dimensionality, which is advantageous given the small size of the real dataset.

Standardization served a dual role. Beyond placing all features on a comparable numerical scale, it interacts favorably with batch normalization and with the Adam optimizer, both of which assume inputs of moderate magnitude for stable behavior. Features with naturally large ranges, such as daily step counts in the synthetic dataset or cholesterol in the real dataset, would otherwise dominate the initial gradient signal and slow convergence. Applying the training-set statistics to the test set without refitting ensured that the evaluation reflected genuine generalization rather than an implicitly adapted transformation.

4.3. Proposed Model

A single fully connected feed-forward network was used throughout. It maps the input vector through three hidden blocks of width 128, 64, and 32 to a two-unit output. Each hidden block combines a linear transformation, batch normalization in the first two blocks, a rectified linear activation, and dropout; the output layer produces class probabilities through a softmax function. The architecture is shown in Figure 2 and specified in Table 4. Applying the identical architecture to both datasets preserves the controlled nature of the comparison; only data-dependent settings such as batch size and epoch count were adjusted for the very different dataset sizes.

5. Mathematical Formulation

This section formalizes the transformations applied by the network, the training objective, and the evaluation metrics. Notation is kept consistent throughout: x denotes an input, W and b denote learnable weights and biases, z denotes a pre-activation, p denotes a predicted probability, and θ denotes the full set of trainable parameters. Feature standardization rescales each input dimension to zero mean and unit variance:

$$x' = \frac{x - \mu}{\sigma} \quad (1)$$

where x is the original feature value, μ and σ are the training-set mean and standard deviation of that feature, and x' is the standardized value used as network input. Standardizing on training statistics alone prevents leakage of test information. Each linear layer applies an affine transformation:

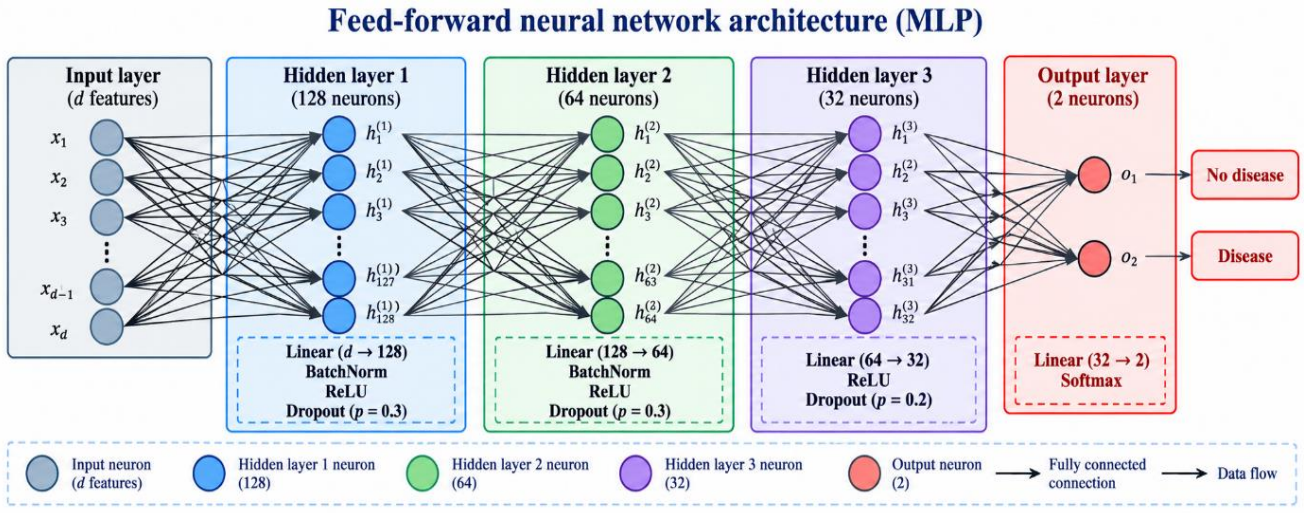


Fig 2: Architecture of the feed-forward network, showing the dimensional flow from input features to the two-class output.

Table 4: Layer specification (d = 25 for synthetic, 13 for real).

#	Layer / Operation	Configuration
1	Linear + BatchNorm + ReLU + Dropout	d → 128, p = 0.3
2	Linear + BatchNorm + ReLU + Dropout	128 → 64, p = 0.3
3	Linear + ReLU + Dropout	64 → 32, p = 0.2
4	Linear (output) + Softmax	32 → 2

$$z = W x + b \quad (2)$$

where x is the layer input, W the weight matrix, b the bias vector, and z the pre-activation output. Stacking such layers with intervening non-linearities yields the network's representational depth. Batch normalization [33] standardizes each pre-activation across a mini-batch and then applies a learnable affine correction:

$$\hat{z} = \frac{z - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}, \quad y = \gamma \hat{z} + \beta \quad (3)$$

where μ and σ² are the per-feature mean and variance computed over the mini-batch, ε is a small stabilizing constant, and γ and β are learnable scale and shift parameters. Normalization stabilizes the activation distribution during training and contributes mild regularization. The rectified linear activation introduces non-linearity:

$$ReLU(z) = max(0, z) \quad (4)$$

which enables the network to model interactions among features; without a non-linearity the composition of linear layers would remain linear. Dropout [34] follows each hidden block, zeroing a fraction p of units during training to discourage co-adaptation. The output layer converts logits to class probabilities through softmax:

$$p_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \quad (5)$$

where z_i is the logit for class i and p_i the predicted probability. The positive-class probability is retained for threshold-based decisions and ROC construction. Training minimizes a class-weighted cross-entropy loss:

$$L = -\sum_i w_i \cdot 1[y = i] \cdot log\, p_i \quad (6)$$

where $1[\cdot]$ is the indicator function, p_i the predicted probability of class i, and w_i the weight for class i. Weights counteract imbalance and are set inversely to class frequency:

$$w_i = \frac{N}{C \cdot n_i} \quad (7)$$

where N is the number of training examples, C the number of classes, and n_i the count in class i. The complete objective adds a weight-decay penalty:

$$L_{total} = L_{CE}(w) + \lambda \|\theta\|^2 \quad (8)$$

where $L_{CE}(w)$ is the weighted cross-entropy of Equation 6, θ the trainable parameters, and λ the weight-decay coefficient. Optimization uses Adam [35], which adapts per-parameter learning rates from running gradient-moment estimates. Performance is assessed with accuracy, the macro-averaged F1 score, and the area under the ROC curve. The F1 score for a class is the harmonic mean of its precision and recall:

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (9)$$

The macro-F1 averages per-class F1 without weighting by prevalence, giving equal importance to both classes. The area under the ROC curve measures threshold-independent discrimination, ranging from 0.5 for random ranking to 1.0 for perfect separation. Reporting the three metrics together separates calibrated classification from underlying ranking ability.

6. Model Implementation and Training Configuration

6.1. Implementation and Environment

The models were implemented in Python using the PyTorch deep learning framework [36], with scikit-learn providing data splitting, preprocessing, imputation, and metric computation [37]. Experiments were executed on the Google Colab platform using a standard CPU runtime, which was sufficient given the compact architecture and modest data volumes. Random seeds were fixed for the framework and for the numerical libraries to promote reproducibility, so that repeated executions yield consistent results. The real dataset was retrieved programmatically through a repository client,

ensuring that the data-loading step is fully scripted rather than dependent on manual file handling.

6.2. Training Configuration

The network was trained with the Adam optimizer using a class-weighted cross-entropy loss. Table 3 lists the hyperparameters. For the large synthetic dataset a batch size of 256 and 80 epochs were used. For the small real dataset a smaller batch size of 32 was adopted, and training was allowed to continue for up to 300 epochs under early stopping. Values examined during the real-data optimization study are indicated in the table.

Table 3: Training hyperparameters in the real-data optimization study.

Hyperparameter	Value
Framework	PyTorch (CPU runtime, Google Colab)
Optimizer	Adam
Initial learning rate	1×10^{-3} (5×10^{-4} searched)
Weight decay (λ)	1×10^{-4} (1×10^{-3} searched)
Loss	Class-weighted cross-entropy
Dropout rates	0.3 / 0.3 / 0.2 (0.5 / 0.5 / 0.3 searched)
Batch size	256 (synthetic) / 32 (real)
Epochs	80 (synthetic) / ≤300 with early stopping (real)
Train / test split	80% / 20%, stratified
Cross-validation	5-fold stratified (real-data study)

6.3. Optimization Protocol for the Real Data

Because the real dataset is small, a single train–test split is a high-variance estimator and an expressive network is prone to overfitting. A separate protocol was therefore applied to the real data and reported apart from the controlled comparison. It combines early stopping on validation loss, five-fold stratified cross-validation, a grid search over learning rate, weight decay, and dropout, and an ensemble that averages the predictions of the fold models under the selected configuration. A hold-out test set was separated before any tuning, confining model selection to the development data. The workflow is shown in Figure 3.

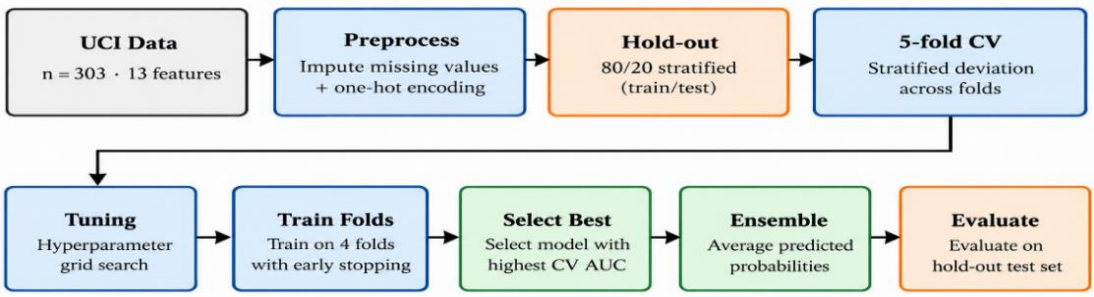


Fig 3: Workflow of the real-data optimization protocol.

7. Result

7.1. Architecture-Controlled Comparison

Table 5 and Figure 7 report performance across the three settings. On the synthetic data the network achieved an accuracy of 0.869, a macro-F1 of 0.850, and a ROC-AUC of 0.940. Transferring the identical architecture to the real data reduced accuracy and macro-F1 to 0.787 while the ROC-AUC remained comparatively high at 0.920. The contrast between a marked accuracy declines and a modest ROC-AUC decline indicates that the model preserved most of its capacity to rank patients by risk on the real data even as its calibrated classification weakened. This is the signature of a decision boundary that is correctly oriented but imprecisely located, consistent with high variance on a small sample.

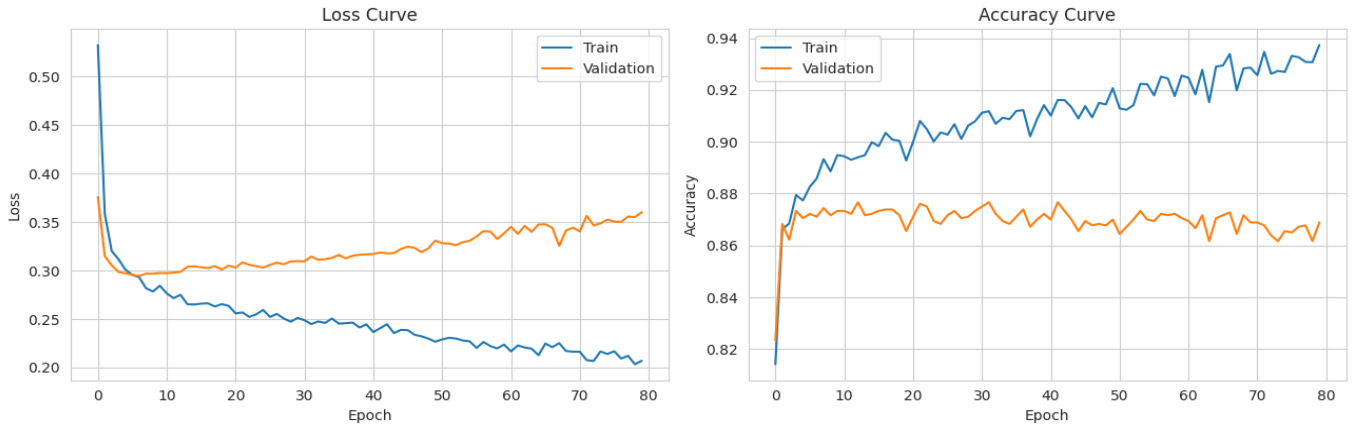


Fig 4: Training and validation loss (left) and accuracy (right) on synthetic dataset across 80 epochs.

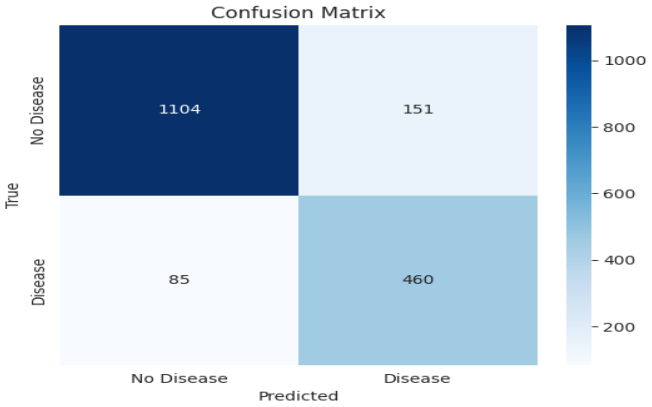


Fig 5: Confusion matrix on the synthetic test set (1,800 patients).

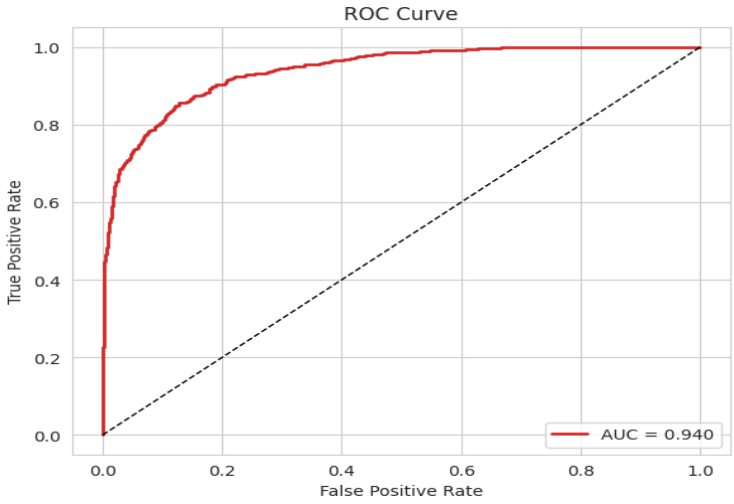


Fig 6: ROC curve on the synthetic test set (AUC = 0.940).

The training curves in Figure 4 illustrate the behavior underlying these numbers. On the synthetic data, training accuracy continued to rise while validation accuracy stabilized near 0.87 and validation loss began to increase after roughly ten epochs, an early indication of overfitting even on the large dataset. The confusion matrix in Figure 5 shows balanced errors, with a modest number of false negatives relative to correctly identified cases, and the ROC curve in Figure 6 confirms strong overall discrimination.

Table 5: Performance across datasets and settings.

Setting	Accuracy	Macro-F1	ROC-AUC
Synthetic (9,000)	0.869	0.850	0.940
UCI real-baseline	0.787	0.787	0.920
UCI real-optimized	0.869	0.869	0.953

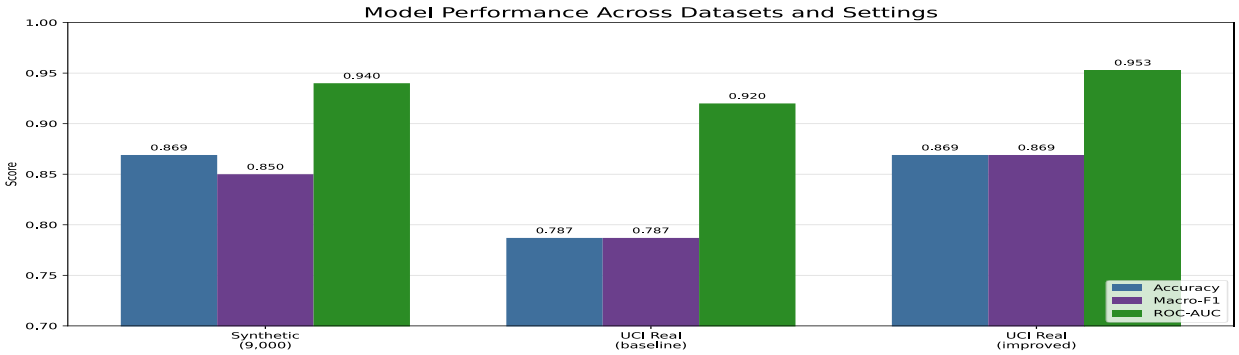


Fig 7: Accuracy, macro-F1, and ROC-AUC across the three settings.

7.2. Real-Data Baseline

When the identical architecture was applied to the real UCI data without the optimization protocol, its behavior differed markedly from the synthetic case. The training curves in Figure 8 show pronounced overfitting: training accuracy approached unity while validation accuracy fluctuated around 0.80 to 0.85, and validation loss rose steadily after the first ten to fifteen epochs. The confusion matrix in Figure 9 reflects the resulting errors on the 61-patient test set, and the ROC curve in Figure 10 records a ROC-AUC of 0.920, indicating that discrimination remained strong despite the weaker calibrated accuracy.

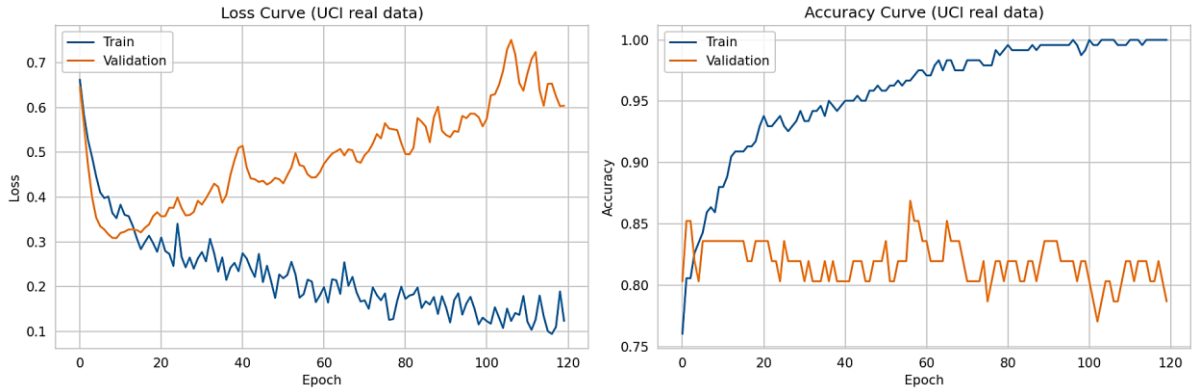


Fig 8: Training and validation loss (left) and accuracy (right) on the real UCI dataset, showing pronounced overfitting.

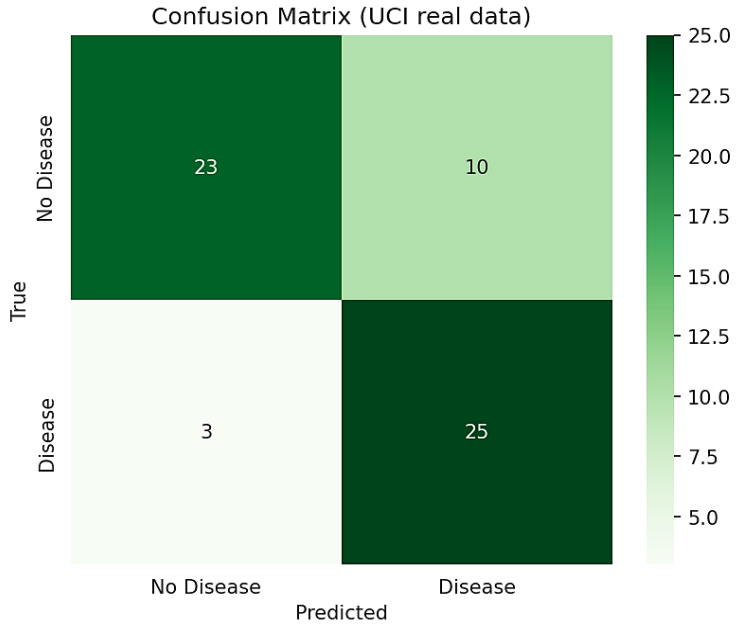


Fig 9: Confusion matrix on the real UCI test set, baseline model (61 patients).

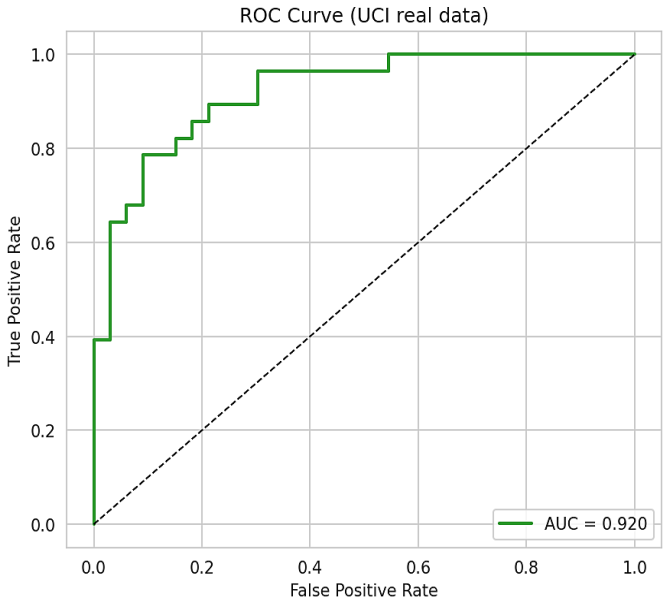


Fig 10: ROC curve on the real UCI test set, baseline model (AUC = 0.920).

7.3. Effect of the Optimization Protocol

The optimization protocol raised real-data accuracy and macro-F1 to 0.869 and ROC-AUC to 0.953, an improvement of roughly eight percentage points in accuracy and more than three points in ROC-AUC relative to the baseline. The optimized real-data model thereby matched the synthetic-data accuracy and exceeded its discrimination. The cross-validation search in Figure 11 shows that the configuration with the strongest dropout achieved the best mean ROC-AUC, direct evidence that overfitting, not an intrinsic property of the real data, limited the baseline. The confusion matrix in Figure 12 and the ROC curve in Figure 13 confirm the improved performance, with fewer false negatives and a smoother, higher ROC curve than the baseline.

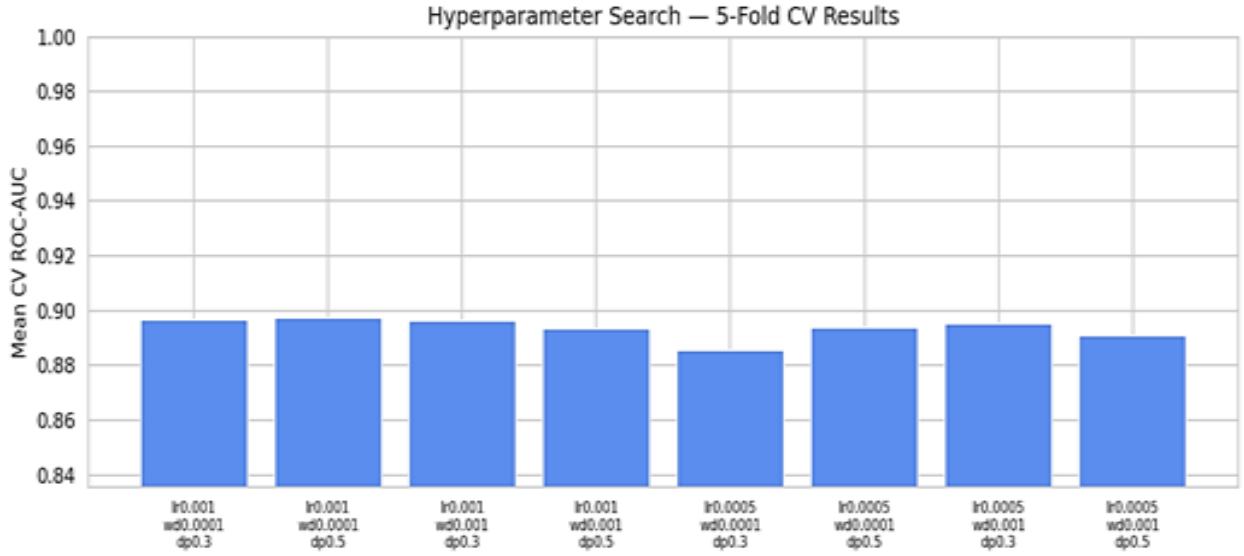


Fig 11: Mean five-fold cross-validation ROC-AUC for each hyperparameter configuration on the real data.

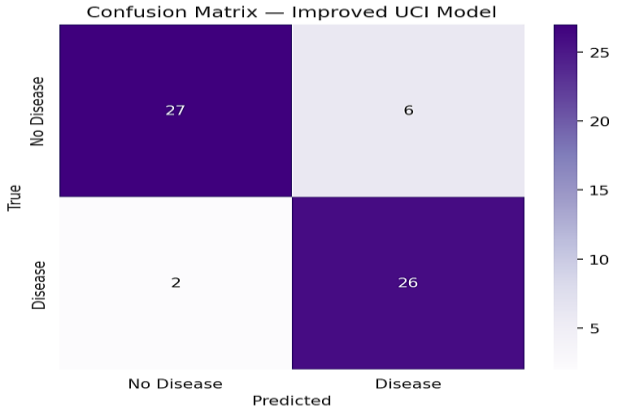


Fig 12: Confusion matrix on the real UCI test set, optimized model.

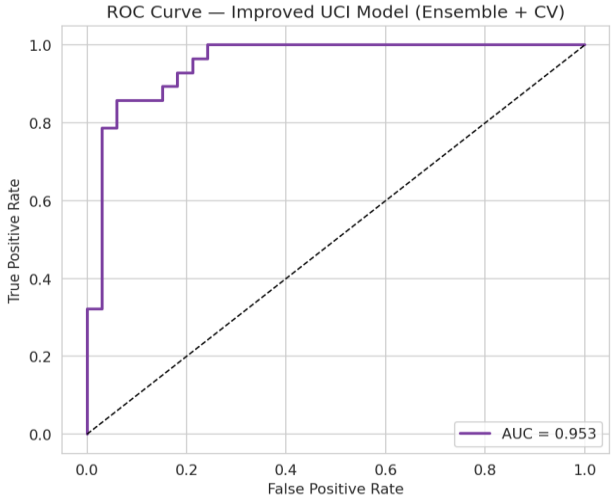


Fig 13: ROC curve on the real UCI test set, optimized model (AUC = 0.953).

8. Validation

Several design elements support the reliability of the reported results, consolidated here.

8.1. Cross-Validation and Held-Out Testing

For the real data, five-fold stratified cross-validation provided a variance-aware estimate during model selection, and a test set separated before tuning provided an unbiased final estimate. The cross-validation standard deviations were small, on the order of 0.02, indicating stable behavior across folds. The agreement between the cross-validation estimate for the selected configuration and the hold-out result supports the absence of selection bias.

8.2. Class Imbalance Handling

Both datasets are imbalanced. The class-weighted loss of Equations 6 and 7 scaled the penalty inversely to class frequency. The balanced per-class precision and recall observed in the results, rather than a collapse toward the majority class, confirm the effectiveness of the weighting.

8.3. Guarding Against Information Leakage

Standardization statistics and imputation values were computed on training data only and then applied to held-out data, and hyperparameter selection was confined to the development partition. These precautions ensure that the test estimates were not inflated by exposure to test information during model construction.

8.4. Partial Ablation of the Optimization Components

The cross-validation search functions as a partial ablation over regularization strength: weaker dropout produced lower mean ROC-AUC and stronger dropout produced the highest, isolating regularization as the decisive factor. The improvement from the single-split baseline to the cross-validated ensemble additionally reflects variance reduction through fold averaging. A full component-wise ablation is identified as future work.

8.5. Stability of the Performance Estimates

The small standard deviations across cross-validation folds provide a quantitative indication that the reported performance is not an artefact of a particular partition. Across the eight hyperparameter configurations examined, mean fold ROC-AUC varied within a narrow band and the fold-level standard deviations remained close to two hundredths, so the ranking of configurations was stable rather than driven by noise. Fixing the random seeds for the framework and numerical libraries further ensured that repeated runs of the same configuration reproduced the same results, which is a prerequisite for meaningful comparison between the baseline and optimized models. Together these properties indicate that the eight-point improvement in accuracy on the real data reflects a genuine effect of the optimization protocol rather than sampling variability.

9. Comparative Analysis

Table 6 places the optimized real-data model alongside representative published studies on the UCI Cleveland benchmark. The obtained accuracy of 0.869 falls within the range reported for well-validated models, roughly 87% to 92%. Figures above this range frequently rely on validation schemes that do not fully separate model selection from testing and are therefore not directly comparable to the strict protocol used here. Within these constraints the model is competitive with established classical and deep learning approaches while using a compact architecture and a transparent procedure. Gradient-boosted tree ensembles such as XGBoost [38] and CatBoost [39] remain strong baselines for tabular clinical data and are natural comparators for future extensions.

Table 6: Comparison with published UCI Cleveland studies.

Study	Method	Accuracy
Mohan *et al.*	Hybrid RF + linear	88.7%
Jindal *et al.*	KNN / LR / RF (mean)	87.5%
Shah *et al.*	KNN	90.8%
Karthick *et al.*	Random forest + selection	88.5%
Demir *et al.*	SVM	92.42%
This work	Optimized MLP (CV + ensemble)	86.9%

Two qualifications apply. First, the cited studies use varied validation protocols, feature-selection strategies, and target encodings, so the accuracies are indicative rather than strictly commensurable. Second, the synthetic-data accuracy should not be entered into such a table as though it were a real-data result, since doing so would conflate two data regimes. The comparison is therefore restricted to the real-data model, and the synthetic result is interpreted separately as a statement about the synthetic task.

10. Discussion

10.1. Principal Findings

Two findings carry the analysis. The architecture transferred from synthetic to real data lost roughly eight points of accuracy while retaining most of its ranking ability, and that loss was reversed once the model was regularized against the small sample. The asymmetry between the two metrics is the more informative of the two observations. Accuracy fell from 0.869 to 0.787 while ROC-AUC moved only from 0.940 to 0.920, which means the ordering of patients by predicted risk survived the change of data regime even though the decision boundary at the default threshold did not. An evaluation reporting accuracy alone would have recorded a substantial failure; an evaluation reporting ROC-AUC alone would have recorded near-equivalence. Neither reading is complete.

10.2. Why Synthetic Performance Exceeded Real Performance

The synthetic corpus presented an easier classification problem than the real cohort. Several mechanisms plausibly contribute, and the evidence here distinguishes between them only partially. Generators that reproduce marginal distributions faithfully may still attenuate higher-order dependencies among variables; when those dependencies carry noise as well as signal in real patients, their attenuation yields cleaner class boundaries and more stable estimates. The pattern observed here, in which discrimination transferred but calibration did not, is consistent with that account. Reduced heterogeneity is a second candidate: a generated cohort contains no measurement error, no missing entries, and no rare presentations, all of which are present in the Cleveland records. The synthetic dataset in this study had no missing values and no duplicates, which is itself a departure from authentic clinical collection.

Related work supports treating statistical fidelity and downstream realism as separate properties. A Bayesian-network study on the same UCI family found that the benefit

of synthetic augmentation tracked the completeness and internal consistency of the source cohort, with the largest gains where the real data were noisiest and smaller, partly non-significant gains elsewhere [26]. At much larger scale, synthetic diabetes records generated by a dual adversarial autoencoder supported models whose predictive performance matched real-data training, yet sex-stratified analysis exposed inconsistencies that aggregate utility metrics did not reveal [27]. Both results point the same way as the present findings: parity on a headline metric does not establish that a synthetic cohort reproduces the structure that matters clinically.

10.3. The Gap Was Remediable, Not Intrinsic

The second finding qualifies the first. After early stopping, five-fold cross-validation, a grid search, and fold ensembling, real-data accuracy reached 0.869 and ROC-AUC 0.953, matching the synthetic accuracy and exceeding its discrimination. The grid search selected the strongest dropout setting, which identifies the operative mechanism directly rather than by inference. What limited the baseline was the interaction between an expressive network and 242 training records, not a property of real clinical data as such. The practical implication is that small real cohorts are not necessarily a barrier to competitive performance, provided the variance introduced by their size is controlled explicitly.

10.4. Implications for Model Development and Validation

These results argue for a staged workflow rather than a substitution. Synthetic data are useful for prototyping architectures, debugging pipelines, and rehearsing analyses under privacy constraints; they are not a substitute for evaluation on real records. A transfer-oriented study of colorectal cancer survival reached the same position from a different direction: a model pretrained on synthetic data and adapted with local hospital data matched a hospital-only baseline across settings, and the authors concluded that synthetic data complement rather than replace real clinical models [28]. Direct zero-shot application in that study reduced performance in most configurations, which is the closest published analogue to the untuned transfer reported here. The architecture-controlled design is what allows these statements to be made with confidence. Studies that vary the model, the feature set, and the evaluation protocol at the same time cannot attribute a performance difference to data provenance. Holding all three fixed and changing only the data makes the attribution direct. The cost of the design is its narrowness: it speaks to one architecture family and two specific cohorts, and the magnitude of the gap reported here should not be read as a general constant.

10.5. Clinical Framing of the Metrics

In screening, the costs of false negatives and false positives are rarely symmetric, and the appropriate operating point reflects that asymmetry rather than a default threshold of one half. ROC-AUC above 0.92 on both datasets indicates that a threshold favouring sensitivity can be selected without sacrificing ranking quality, and the softmax outputs make that adjustment straightforward. Reporting threshold-independent discrimination alongside a fixed-threshold accuracy therefore conveys more usable information than accuracy alone, particularly when a model developed under one data regime is to be deployed under the cost structure of another.

11. Limitations and Future Work

- The two datasets differ in feature set as well as size, so the comparison reflects data provenance in aggregate rather than a matched feature-by-feature correspondence. Constructing aligned synthetic and real cohorts would isolate the effect more sharply.
- One generator and one architecture family were examined. Whether the transfer gap behaves similarly for adversarial or diffusion-based generators, or for tree ensembles and tabular transformers, is untested here.
- The real dataset contains 303 records and test partition 61. Estimates on a partition of that size carry appreciable uncertainty even with cross-validation and ensembling; external validation on larger multi-site cohorts is required before any clinical reading.
- Feature attribution was not performed [40], so the study does not establish whether the synthetic and real models rely on the same predictors. This is a natural next step and would test the distribution-shift explanations offered in Section 10.2.
- The ablation of the optimization protocol was partial. Future work should include a component-wise ablation study to quantify the individual contributions of early stopping, cross-validation, and ensembling.

12. Conclusion

A single feed-forward network applied without modification to a synthetic cohort of 9,000 records and to the 303-patient UCI Cleveland dataset achieved 0.869 accuracy and 0.940 ROC-AUC on the former and 0.787 accuracy and 0.920 ROC-AUC on the latter. The divergence between the two metrics indicates that ranking ability transferred while calibrated classification did not. A separate optimization protocol on the real data, combining early stopping, five-fold cross-validation, hyperparameter search, and fold ensembling, raised real-data performance to 0.869 accuracy and 0.953 ROC-AUC, with the search selecting the strongest regularization setting. The gap was therefore a consequence of fitting an expressive model to a small sample, not a property of the real data.

Two claims follow from this evidence, and no more. Performance measured on synthetic data can overstate the difficulty of the corresponding real task, so synthetic results require confirmation on real records before clinical interpretation. And competitive performance on small real cohorts is attainable through explicit variance control, without additional data. The architecture-controlled design supports these claims for one model family and two cohorts; extending them to other generators, architectures, and clinical domains remains open.

Funding

No funding was received for this manuscript.

Conflicts of Interest

The authors declare no conflicts of interest.

References

1. Roth GA, Mensah GA, Johnson CO, *et al*. Global burden of cardiovascular diseases and risk factors, 1990-2019. Journal of the American College of Cardiology. 2020;76(25):2982-3021. https://doi.org/10.1016/j.jacc.2020.11.010.

2. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521:436-444.https://doi.org/10.1038/nature14539.
3. Rajkomar A, Dean J, Kohane I. Machine learning in medicine. New England Journal of Medicine. 2019;380(14):1347-1358. https://doi.org/10.1056/NEJMra1814259.
4. Ansari GA, Bhat SS, Ansari MD, Ahmad S, Nazeer J, Eljialy AEM. Performance evaluation of machine learning techniques (MLT) for heart disease prediction. Computational and Mathematical Methods in Medicine. 2023;2023:8191261. https://doi.org/10.1155/2023/8191261.
5. Chicco D, Jurman G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics. 2020;21:6. https://doi.org/10.1186/s12864-019-6413-7.
6. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial networks. Communications of the ACM. 2020;63(11):139-144. https://doi.org/10.1145/3422622.
7. Kingma DP, Welling M. Auto-encoding variational Bayes. Proceedings of the International Conference on Learning Representations (ICLR); 2014. https://doi.org/10.48550/arXiv.1312.6114.
8. Xu L, Skoularidou M, Cuesta-Infante A, Veeramachaneni K. Modeling tabular data using conditional GAN. Advances in Neural Information Processing Systems. 2019;32:7335-7345. https://doi.org/10.48550/arXiv.1907.00503.
9. Mohan S, Thirumalai C, Srivastava G. Effective heart disease prediction using hybrid machine learning techniques. IEEE Access. 2019;7:81542-81554. https://doi.org/10.1109/ACCESS.2019.2923707.
10. Jindal H, Agrawal S, Khera R, Jain R, Nagrath P. Heart disease prediction using machine learning algorithms. IOP Conference Series: Materials Science and Engineering. 2021;1022:012072. https://doi.org/10.1088/1757-899X/1022/1/012072.
11. Shah D, Patel S, Bharti SK. Heart disease prediction using machine learning techniques. SN Computer Science. 2020;1:345. https://doi.org/10.1007/s42979-020-00365-y.
12. Karthick K, Aruna SK, Samikannu R, Kuppusamy R, Teekaraman Y, Thelkar AR. Implementation of a heart disease risk prediction model using machine learning. Computational and Mathematical Methods in Medicine. 2022;2022:6517716. https://doi.org/10.1155/2022/6517716.
13. Bharti R, Khamparia A, Shabaz M, Dhiman G, Pande S, Singh P. Prediction of heart disease using a combination of machine learning and deep learning. Computational Intelligence and Neuroscience. 2021;2021:8387680. https://doi.org/10.1155/2021/8387680.
14. Demir S, Selvitopi H, Selvitopi Z. An early and accurate diagnosis and detection of coronary heart disease using deep learning and machine learning algorithms. Journal of Big Data. 2025;12:228. https://doi.org/10.1186/s40537-025-01283-7.
15. Zhou C, Dai P, Hou A, *et al*. A comprehensive review of deep learning-based models for heart disease prediction. Artificial Intelligence Review. 2024;57:263. https://doi.org/10.1007/s10462-024-10899-9.
16. García-Ordás MT, Bayón-Gutiérrez M, Benavides C, Aveleira-Mata J, Benítez-Andrades JA. Heart disease risk prediction using deep learning techniques with feature augmentation. Multimedia Tools and Applications. 2023;82:31759-31773. https://doi.org/10.1007/s11042-023-14817-z.
17. Ali F, El-Sappagh S, Islam SMR, Kwak D, Ali A, Imran M, Kwak KS. A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion. Information Fusion. 2020;63:208-222. https://doi.org/10.1016/j.inffus.2020.06.008.
18. Rathod VU, *et al*. A modular deep learning architecture for interpretable disease prediction across tabular clinical and biometric datasets. PLOS ONE. 2026. https://doi.org/10.1371/journal.pone.0348670.
19. Hashan AM, Saha S, Al-Saeedi AAK, Md Tariqur Rahman S, Mily A, Karim MN. Machine learning and hyperparameter optimization for liver disease prediction in organizational systems. Proceedings of the IEEE Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT), IEEE; 2025:69-72. https://doi.org/10.1109/USBEREIT65494.2025.11054156.
20. Hashan AM, Agbozo E, Al-Saeedi AAK, Saha S, Haidari A, Rabi MNF. Brain tumor detection in MRI images using image processing techniques. Proceedings of the 4th International Symposium on Agents, Multi-Agent Systems and Robotics (ISAMSR), IEEE; 2021:24-28. https://doi.org/10.1109/ISAMSR53229.2021.9567799.
21. Gonzales A, Guruswamy G, Smith SR. Synthetic data in health care: A narrative review. PLOS Digital Health. 2023;2(1):e0000082. https://doi.org/10.1371/journal.pdig.0000082.
22. Pezoulas VC, Zaridis DI, Mylona E, Androutsos C, Apostolidis K, Tachos NS, Fotiadis DI. Synthetic data generation methods in healthcare: A review on open-source tools and methods. Computational and Structural Biotechnology Journal. 2024;23:2892-2910. https://doi.org/10.1016/j.csbj.2024.07.005.
23. Murtaza H, Ahmed M, Khan NF, Murtaza G, Zafar S, Bano A. Synthetic data generation: State of the art in health care domain. Computer Science Review. 2023;48:100546. https://doi.org/10.1016/j.cosrev.2023.100546.
24. Hernandez M, Epelde G, Alberdi A, Cilla R, Rankin D. Synthetic data generation for tabular health records: A systematic review. Neurocomputing. 2022;493:28-45. https://doi.org/10.1016/j.neucom.2022.04.053.
25. Foraker RE, Yu SC, Gupta A, Michelson AP, Pineda Soto JA, Colvin R, Loh F, Kollef MH, Maddox TM, Payne PRO. Spot the difference: Comparing results of analyses from real patient data and synthetic derivatives. JAMIA Open. 2020;3(4):557-566. https://doi.org/10.1093/jamiaopen/ooaa060.
26. Lazzaro I, *et al*. Assessing the impact of synthetic data generated by Bayesian networks on heart disease prediction. Journal of Computational Science. 2026.
27. Ortuño F, de la Oliva Roque VM, Ramirez-Lopez JI, Kreil DP, Dopazo J, Loucera C. High-fidelity synthetic data replicates clinical prediction performance in a million-patient diabetes cohort. Advanced Science.

2026;13(29):e16196. https://doi.org/10.1002/advs.202516196.
28. Jang Y, Kwon JH, Kim H, Joung YJ, Nang J, Kim CH. Evaluation of the applicability of synthetic data in the development of colorectal cancer survival prediction models: External validation of advanced machine learning models based on National Cancer Data Center data. Journal of Medical Internet Research. 2026. https://www.jmir.org/2026/1/e86087.
29. Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural Networks. 1989;2(5):359-366. https://doi.org/10.1016/0893-6080(89)90020-8.
30. Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: Data mining, inference, and prediction. 2nd ed. Springer; 2009. https://doi.org/10.1007/978-0-387-84858-7.
31. Detrano R, Janosi A, Steinbrunn W, Pfisterer M, Schmid JJ, Sandhu S, Guppy KH, Lee S, Froelicher V. International application of a new probability algorithm for the diagnosis of coronary artery disease. American Journal of Cardiology. 1989;64(5):304-310. https://doi.org/10.1016/0002-9149(89)90524-9.
32. Dua D, Graff C. UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences; 2019. http://archive.ics.uci.edu/ml.
33. Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. Proceedings of the 32nd International Conference on Machine Learning; 2015:448-456. https://doi.org/10.48550/arXiv.1502.03167.
34. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research. 2014;15(1):1929-1958.
35. Kingma DP, Ba J. Adam: A method for stochastic optimization. Proceedings of the International Conference on Learning Representations (ICLR); 2015. https://doi.org/10.48550/arXiv.1412.6980.
36. Paszke A, Gross S, Massa F, *et al*. PyTorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems. 2019;32:8024-8035. https://doi.org/10.48550/arXiv.1912.01703.
37. Pedregosa F, Varoquaux G, Gramfort A, *et al*. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research. 2011;12:2825-2830. https://doi.org/10.48550/arXiv.1201.0490.
38. Chen T, Guestrin C. XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2016:785-794. https://doi.org/10.1145/2939672.2939785.
39. Prokhorenkova L, Gusev G, Vorobev A, Dorogush AV, Gulin A. CatBoost: Unbiased boosting with categorical features. Advances in Neural Information Processing Systems. 2018;31:6638-6648. https://doi.org/10.48550/arXiv.1706.09516.
40. Lundberg SM, Lee SI. A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems. 2017;30:4765-4774.

How to Cite This Article
Al-Maamori YH. Assessing the transferability of deep learning from synthetic to real clinical data for heart disease prediction: an Architecture-Controlled comparative study. Int J Future Eng Innov. 2026;3(6):31-43. doi:10.54660/IJFEI.2026.3.6.31-43.

Creative Commons (CC) License
This is an open access journal, and articles are distributed under the terms of the Creative Commons Attribution NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) License, which allows others to remix, tweak, and build upon the work non-commercially, as long as appropriate credit is given and the new creations are licensed under the identical terms