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: 22-8-2026
Accepted: 24-9-2026
Published: 26-10-2026
Page No: 31-43

Abstract
Where real clinical information is limited, synthetic patient records are becoming more popular, Limited due to privacy, or too costly to label. How far performance on such data to predict performance on authentic records for tabular cardiac is still unclear. prediction. This study is a fixed architecture study where only the data is changed. One feed-forward A synthetic corpus of 9,000 patient texts was used to train network, which was maintained in the same structure. under both the simulated UCI Cleveland Heart Disease database (303 patients), The same preprocessing, optimization based on classes and evaluation has been used, so that any difference only indicates data provenance; it is not a reflection of model design. On the synthetic data the accuracy and AUC of network were 0.869 and 0.940 respectively. Sent to the real data, accuracy had fallen to 0.787 while AUC remained at 0.920: the model was still the best ranked. It performed well for the patients, but its calibrated classification was degraded. A second experiment on the real data used early stopping, five-fold stratified cross validation, a hyperparameter, grid search and fold level ensemble. The accuracy was restored to 0.869 and AUC increased to This is in line with the published results on the same benchmark (0.953). The key words that were chosen for the search are the most strongly regularized configuration (overfitting on a small sample). This is not an inherent difficulty of the actual data, but rather a consequence of the differences between the real data and the simulated data. There are two practical implications: Synthetic data can make a classification task look easier than it is, and the gap this Standard regularization is fairly closed for creates. Synthetic datasets are applicable to model prototyping — but results from these models need to be tested with model validation. real clinical data prior to clinical interpretation.

DOI: http://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 (CVD) continues to be a major cause of death and disability worldwide and has a continuous demand upon preventive and curative health services [1]. Early identification of those who are at increased risk prior to the occurrence of acute events provide the opportunity to intervene early, and to allocate limited clinical resources. For decades, there has been a focus on computational risk prediction, beginning with regression-based scores that use limited, prespecified variables, and moving on to machine learning-based scores and, in recent years, those based on deep learning models which can represent complex, non-linear relationships among numerous variables [2, 3, 4, 5].

Any data-driven cardiac model is constrained by the data on which it is based. Real clinical data are often small, imbalanced, incompletely labeled, and hard to share, due to institutional governance and privacy regulation. Such restrictions have led to a growing curiosity about synthetic data—synthetic records which mimic the statistical characteristics of real cohorts but by

artificially generated to minimize direct exposure of patient information. Now widely available for teaching, prototyping, and benchmarking, generation can be achieved through simple marginal sampling methods to generative adversarial networks [6], variational autoencoders [7] and conditional models for tabular datasets [8].

This dependence raises a methodological issue that has been little explored from an empirical perspective in the cardio tabular context. This does not imply that a model that performs well on a synthetic dataset will perform well on real patients, nor that the same reported metrics will apply in both scenarios. A generator can have more regular dependencies, which are easier to classify than the ones seen in the clinic, leading to an optimistic estimate of performance; or it can have artefacts that reduce performance. Without a controlled comparison, it is unknown whether the findings of synthetic data can be equated with real data, and conclusions drawn from synthetic data alone, may not accurately capture clinical readiness. The existing gap in literature is very specific. A large body of work has been devoted to statistical fidelity assessment, comparing the marginal and joint distributions of synthetic data with cohorts of real data, and a separate body of work has been devoted to predictive performance assessment, reporting results of deep models on real cardiac data. A few studies have attempted to tie these strands together by keeping a fixed predictive model and testing the performance of the model by changing the data provenance alone. Few studies have attempted to connect these strands by testifying the performance of a predictive model with a fixed one and varying the data provenance only. When studies do compare synthetic and real data, it is typically done so in a manner that the model, the features studied, and the way the result is assessed are all modified, obscuring the effect of data provenance from modeling decisions. To confidently attribute performance differences to provenance, an experiment with the same protocol and architecture, but varying data is necessary. The main goal of this study is to provide such an experiment for cardiac tabular prediction. This study has four contributions. First, it offers an architecture-controlled comparison of synthetic and real clinical data for heart disease prediction, eliminating model choice as a confounding factor. Second, it provides complementary measures of the transfer gap, and demonstrates that data provenance has differential effects on discrimination and calibrated accuracy. Third, it shows, by means of a controlled optimization study, that the gap is caused by overfitting on the relatively small sample of real data, which can be overcome using standard methods. Fourth, it places the results in the published benchmarks for UCI, not allowing it to be confused with the real data performance.

2. Literature Review

2.1. Machine and Deep Learning for Cardiac Prediction

For over 30 years, the UCI Cleveland cohort has been a reference point for cardiac classification and the overall results have generated a valuable expectation range. A hybrid model suggested by combining random forest with linear learner achieved accuracy in the vicinity of 88.7%, saying that the hybrid of complementary learners has a mixture of axis-aligned and linear structure [9].

The mean accuracy were roughly 87.5% for the evaluation of K-nearest neighbours, logistic regression, and random forests, and showed sensitivity to feature selection [10]. 90.8% [11] was obtained by A K-nearest-neighbour classifier and 88.5% [12] for random forests with feature selection. 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. Over the past five years (2018-2023), the trend in comprehensive reviews is moving from single feed forward networks to integrated architectures with deep models coupled to feature selection, optimization or attention [15]. In one study, the cardiac features were fed into a sparse autoencoder prior to classification, similar to the regularization philosophy of the present work [16] and boosting the precision on small datasets. Another ensemble of combined DL and feature fusion in both Cleveland and Hungarian datasets, achieved high accuracy, but needed more sophisticated data manipulation which makes comparison difficult [17]. Recent studies have also introduced modular pipelines that choose between multilayer perceptrons, one-dimensional convolutional networks and tabular transformers based on the characteristics of the dataset, achieving an area under the curve of almost 0.90 on the same benchmark of 303 records from the Cleveland heart disease dataset under nested cross-validation [18]. These studies report accuracy in the high 80s to mid-90s range in these studies, and the high numbers are often accompanied by carefully designed feature engineering or validation methods which are not strictly separated from model selection to test.

These methods are not only relevant to cardiac data for prediction, but to clinical tabular prediction in general. The liver disease prediction task has been used as a benchmark for machine learning with systematic hyperparameter optimization, and the results have shown that tuning and ensemble tree models have a significant impact on predictive accuracy on tabular medical records [19]. The result is similar to that of the current work, where hyperparameter tuning and ensembling were key factors for success in the small real data set. The imaging domain is a good example of the wide range of applications in which deep convolutional networks have been used to solve diagnostic problems like brain tumour detection in magnetic resonance imaging [20]. This shows a broad range of deep learning in the healthcare field and highlights the fact that there are different modelling challenges for each modality: tabular and image. The present work falls under the Tabular domain where sample size and regularization are dominant factors to influence performance rather than feature extraction from the spatial domain.

One methodological warning that occurs throughout this literature is that small data sets can lead to positive evaluation. If the same data that is used for final reporting are used to tune the hyperparameter or there is no untouched hold-out set, accuracy estimates can be inflated. This is why several authors do suggest to cross validate, with explicit separation of tuning and testing. The protocol used in this work is based on that recommendation and reported values are considered to be estimates, not best values.

2.2. Synthetic Clinical Data and Its Validation

The field of synthetic data generation is now a separate research branch in the context of healthcare informatics. Some reviews follow this trail from the marginal-based methods to generative adversarial networks, variational autoencoders and diffusion models, and highlight that the usefulness of synthetic data is not just about the similarity of the statistics but relies on the downstream utility as well [21], [22], [23]. A recent series of systematic reviews of tabular health-record synthesis reach comparable results about the priority 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 learning study of survival in colorectal cancer came to a similar conclusion: A model trained on synthetic data and locally adapted with hospital data performed as well as a model trained on the hospital data alone, but the authors noted that synthetic data should be used in conjunction with real clinical data, not in place of it. [28] Although these advances have been made, relatively little research has been conducted with a fixed predictive model using synthetic and real clinical data, and comparing the difference. The present study makes such a controlled comparison for cardiac tabular prediction and connects to the real-life issue of overfitting on small datasets. A summary of the representative studies and their comparisons to the present work are presented in Table 1.

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 formulated as a supervised binary classification problem. The goal is to find a function that maps a feature vector to a probability of whether the patient has heart disease (or not) from a set of labelled examples, and that works with new feature vectors that have not been seen before.

3.1. Supervised Classification Framework

Make each patient correspond to a feature vector, and a label from an unknown joint distribution over features and outcomes. Classifier is a function with parameters that approximates the probability of the disease when provided with the features. Learning involves choosing parameters that have a minimum discrepancy with respect to a measure between the probability functions predicted by the parameters and the finite sample of labels, with the assumption that the same set of parameters will be appropriate when confronted with independent data from the same distribution. The challenge is generalization — models that are sufficiently complex can fit the training sample well and still perform poorly on new instances, even more so as the number of instances in the training set decreases compared to the number of model parameters.

3.2. Feed-Forward Neural Networks As Universal Approximators

Alternating linear transformations and non-linear activation functions form a feed-forward neural network. If there is at least one hidden layer, and the activation function is non-linear, then the networks are universal approximators which can represent any continuous function on a compact domain to arbitrarily high accuracy when the width of the network is large enough [29]. This expressive ability encourages their use for clinical prediction, in which there are non-linear and interacting relationships between risk factors. It is the same ability that makes regularization explicit when the training sample is small, in that an unconstrained network can learn idiosyncratic patterns that don't generalize. Later in the paper methods are developed specifically to control this capacity.

3.3. The Bias–Variance Perspective on Small Clinical Datasets

The behavior in this study can be explained very well in terms of the bias–variance decomposition of generalization error [30]. A flexible model with a large regular data set can have both a low bias and a low variance. The same model tends to have high variance when applied to a small data set: predictions will change substantially if the model is trained on a different data set, and its decision boundary, well-

aligned with the class boundaries, may be poorly placed. Regularization methods like dropout, weight decay, early stopping and ensembling minimize the variance, resulting in a slight increase in bias, and thus in better expected accuracy on unseen data. This view puts the empirical comparison of the large synthetic data set with the small real data set into perspective and sheds light on why the optimization protocol is able to bridge the gap observed.

4. Experimental Setup

4.1. Dataset Preparation

For this study two different datasets with opposing origin and size were chosen (Table 2). The synthetic data set is a publicly available set of 9,000 synthetic samples of the Kaggle corpus, with an estimated disease prevalence of 30%, as described by the authors. For each record there are 25 predictive attributes including 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)

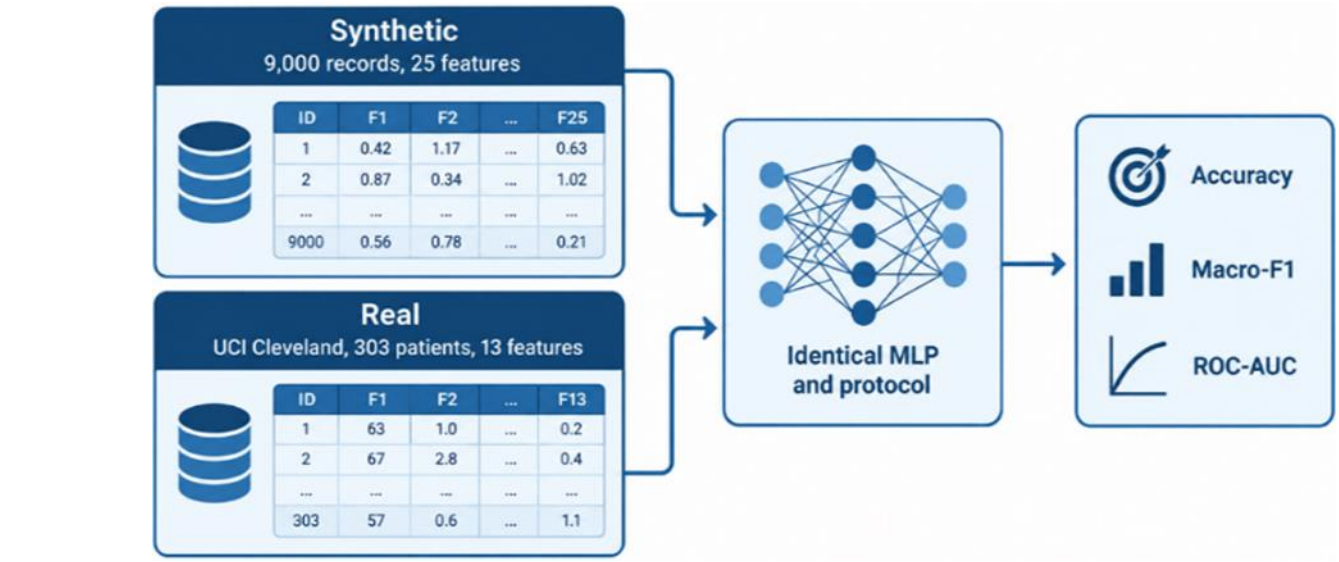


Fig 1: Overview of the two datasets feeding a single shared model.

4.2. Preprocessing

For both data sets, the same pre-processing was used. Examples of non-numeric attributes were label encoded. For the real dataset, missing values were replaced by the median values of the features, which is a suitable value for skewed clinical variables. Data for each set were split into a training and test set, with a 20% test set, and features were standardized based on statistics calculated on the training partition only. The formal definition of standardization is given in Section 5.

The median value is used instead of the mean value because of the distribution of the variables concerned. The real data set's vessel-count and thalassemia fields are discrete and asymmetric, and the median provides a realistic central value, which is not adversely affected by extreme values. The imputation procedure had limited influence on the overall distributions of the features, due to the fact that only a small proportion of records needed imputation, and the imputation procedure did not use information from the test partition. The categorical fields were encoded using label encoding rather than one-hot encoding to reduce the dimensionality of the input space, especially since the first linear layer in the network can handle an ordinal encoding afterwards without a significant increase in its size, which is beneficial since the real data is small.

The standardization was done for two purposes. It serves many more purposes than just putting all features on a similar numerical scale: It works well with batch normalization and with the Adam optimizer, which require moderate-sized features to behave well. In the synthetic data, a large range of features (such as the step numbers per day) would otherwise swamp the starting gradient signal and slow down convergence, and in the real data, one feature (cholesterol) has a large range. The test set was used without retraining, which guaranteed that the test was truly a generalization and not some transformation that was simply adapted from the training set.

4.3. Proposed Model

All along, a single fully interconnected feed-forward network was used. It applies 3 hidden layers of sizes 128, 64, and 32 to the input vector and outputs 2. The hidden blocks are linear transformations, batch normalization (in the first two blocks), rectified linear activations and dropout; the output layer generates class probabilities via a softmax layer. Architecture is displayed in Figure 2, and specified in Table 3. The same architecture was used for both datasets, and only data-dependent parameters like the batch size and number of epochs were altered for the significantly different data sizes.

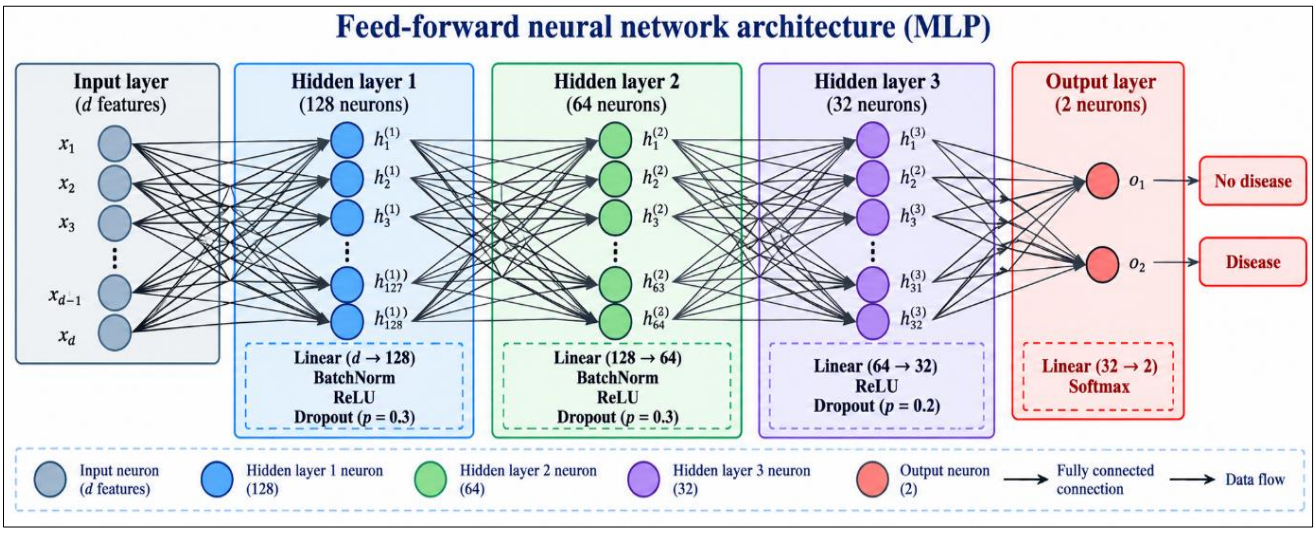


Fig 2: Architecture of the feed-forward network, showing the dimensional flow from input features to the two-class output.

Table 3: 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

5. Mathematical Formulation

This section makes the transformations applied by the network, the training objective and the evaluation metrics formal. 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:

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

Where x is the feature's value as presented in the training set, μ is the mean of the training-set's feature values, and σ is the standard deviation of its feature values. Where x is the value of the original feature, μ represents the mean value of the feature in the training set, and σ represents the standard deviation of the feature values in the training set, and x' is the value of the standardized feature as it will be used as network input. When training with only statistics, the leakage of information from the test is prevented. Each linear layer applies an affine transformation:

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

where x is the layer input, W the weight matrix, b the bias vector, and z the pre-activation output. The network's representational depth is obtained by chaining such layers together with intervening non-linearities. The idea of batch normalization [33] is to normalize the values of each pre-activation within a mini-batch, followed by a learnable affine correction:

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

where μ is the per-feature mean and σ² is the variance calculated from the mini-batch, and ε is a small value added for stability, while γ and β are the parameters that will be learned, which are used for scaling and shifting. It helps to make the activation distribution more stable during training and provides mild regularization. The rectified linear activation introduces non-linearity:

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

This allows the network to learn interactions between the features, otherwise the composition of linear layers would be linear. Dropout [34] is applied after every hidden block of units and causes the fraction (p) of the units to be turned off during training to prevent co-adaptation. We use softmax to map logs to class probabilities in the output layer:

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

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:

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

Where 1[•] is the indicator function, p_i the predicted probability of class i, and w_i the weight for class i. Weights are used to reduce imbalance and they are inversely proportional to the class frequency:

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

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:

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

where $L_{CE}(w)$ is the weighted cross-entropy of Equation 6, and λ the weight-decay coefficient. Optimization is via Adam [35] which learns per-parameter learning rates based on running gradient-moment estimates. The accuracy, macro-averaged F1 score and area under the ROC curve are

used to measure performance. F1 score for a class is the harmonic average of its precision and recall:

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

The macro-F1 is the average of per-class F1s without prevalence weighting, a weighting which assigns equal importance to both classes. The area under the ROC curve is threshold independent, from 0.5 (random ranking) to 1.0 (perfect separation). Reporting the three metrics together removes the influence of calibrated classification from underlying ability to rank.

6. Model Implementation and Training Configuration

6.1. Implementation and environment

Implementations were realized in Python with the PyTorch deep learning framework [36] and with the data splitting, data preprocessing, imputation, and computation of the metrics using scikit-learn [37]. In the case of the compact architecture and relatively small data set, the standard CPU runtime was used for the experiments, which ran on the Google Colab platform. Random seeds were set for the framework and for the numerical libraries for better reproducibility, i.e. repeated runs should give the same result. The real data was obtained programmatically via a repository client, so that the loading of data is fully scripted and not reliant on manual handling of files.

6.2. Training configuration

A class weighted cross-entropy loss was used to train the network with the Adam optimizer. The hyperparameters are given in Table 4. A batch size of 256 and 80 epochs were used for the big synthetic data set. A smaller batch size of 32 was used for the small real dataset and early stopping was used to allow the training to proceed up to 300 epochs. Values used in the real-data optimization study are marked in the table.

Table 4: 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

The real dataset is small, and the train–test split is a high variance estimator, and the expressive network is likely to over fit the data. For the real data, a separate protocol was then used, and is reported separately from the controlled comparison. It includes early stopping on validation loss, five-fold stratified cross-validation, a grid search on learning rate, weight decay and dropout and an average model that takes the results from each of the folds under one of the best parameters. No tuning was performed and a hold-out test set was separated, thus limiting model selection to the development data. The workflow is shown in Figure 3.

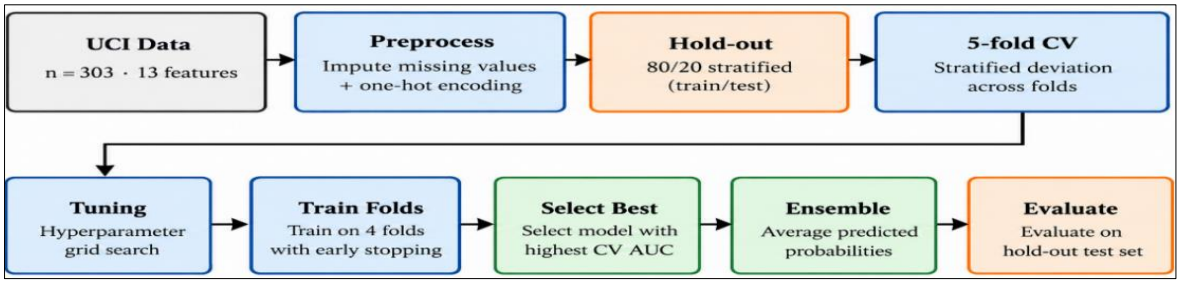


Fig 3: Workflow of the real-data optimization protocol.

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.

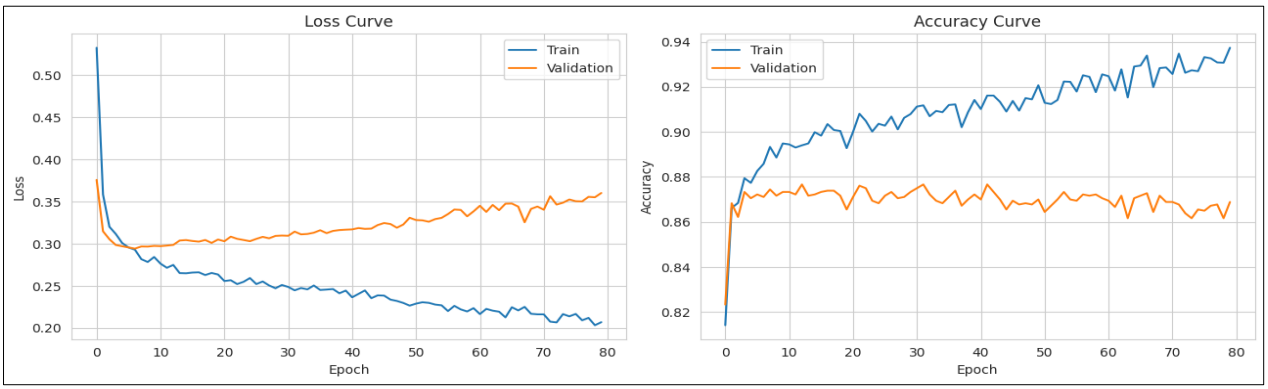


Fig 4: Training and validation loss (left) and accuracy (right) on synthetic dataset across 80 epochs.

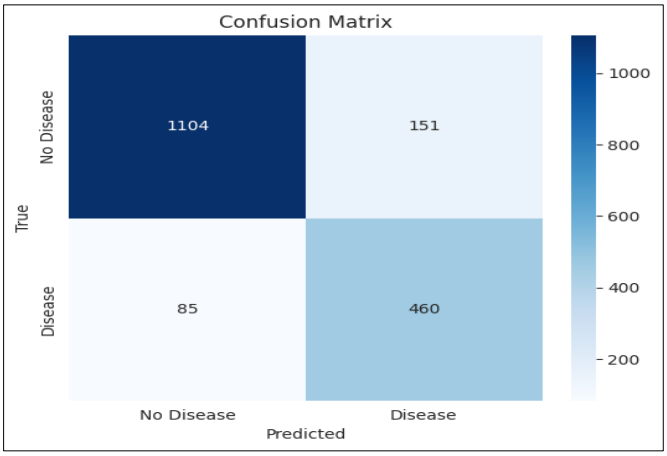


Fig 5: Confusion matrix on the synthetic test set (1,800 patients).

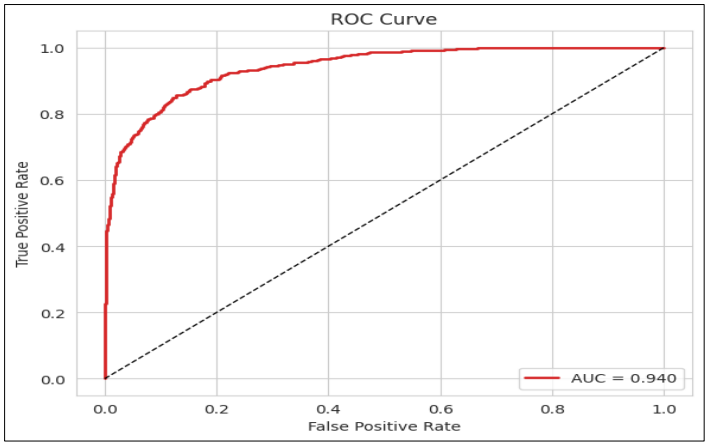


Fig 6: ROC curve on the synthetic test set (AUC = 0.940).

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

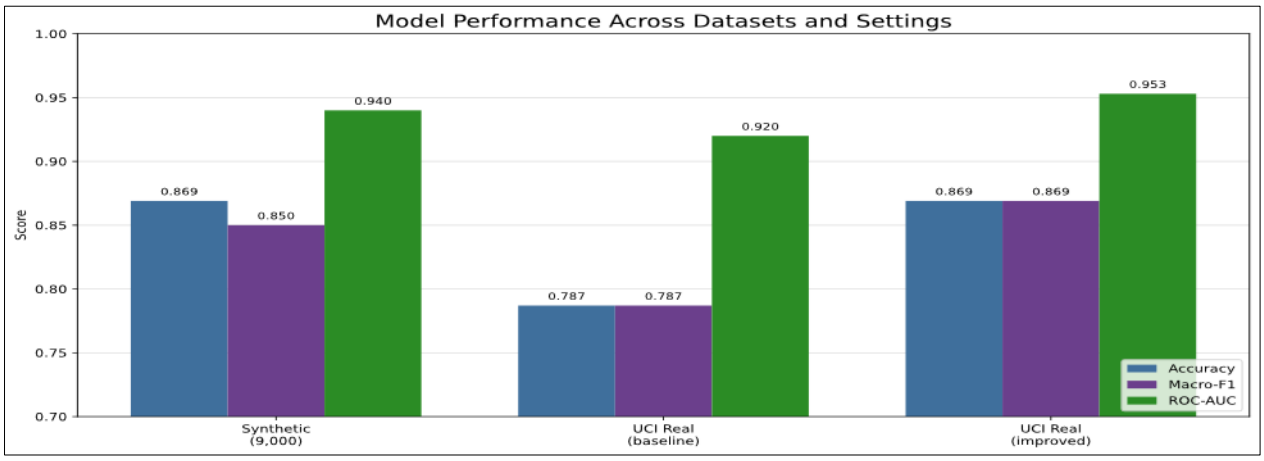


Fig 7: Accuracy, macro-F1, and ROC-AUC across the three settings.

7.2. Real-data baseline

If the same architecture was used with the actual UCI data, but without the optimization protocol, its behavior would be significantly different from the synthetic case. The curves in the training plot of Figure 8 illustrates significant overfitting, with the training accuracy reaching near 1 and the validation accuracy oscillating between 0.80 and 0.85 with increasing loss with the number of 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.

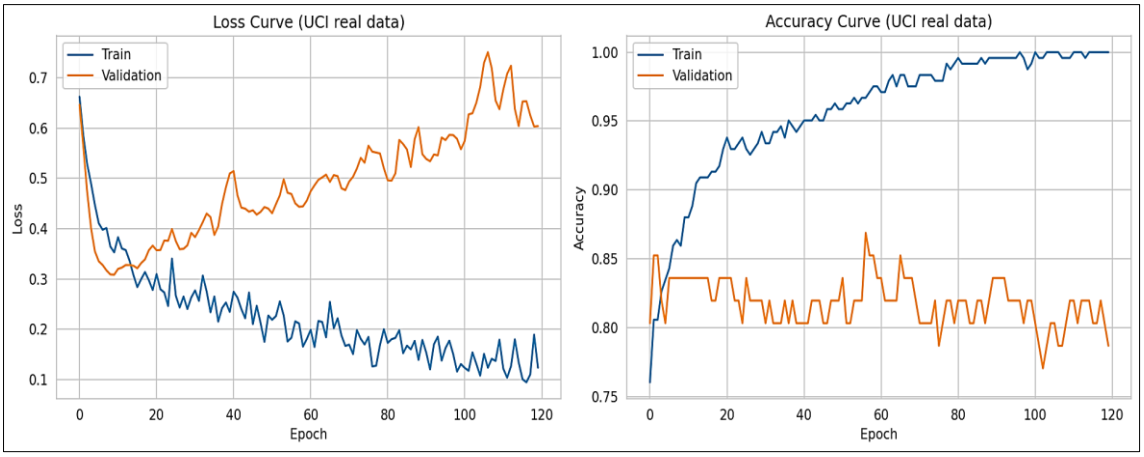


Fig 8: Training and validation loss (left) and accuracy (right) on the real UCI dataset, showing pronounced overfitting.

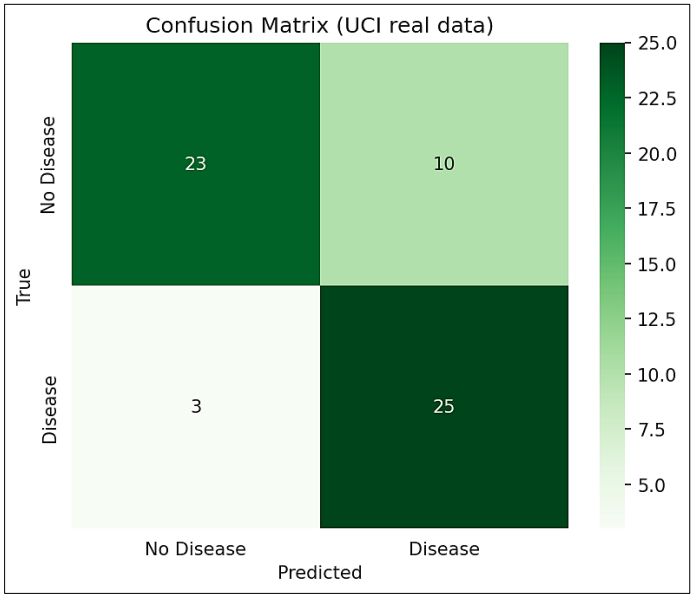


Fig 9: Confusion matrix on the real UCI test set, baseline model (61 patients).

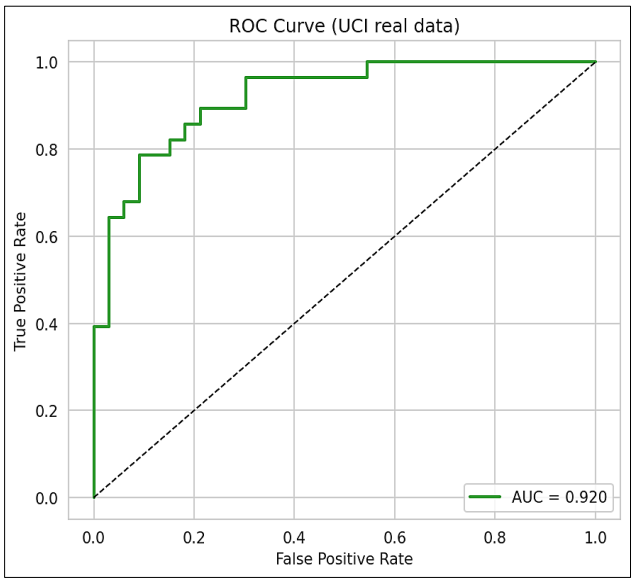


Fig 10: ROC curve on the real UCI test set, baseline model (AUC = 0.920).

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.

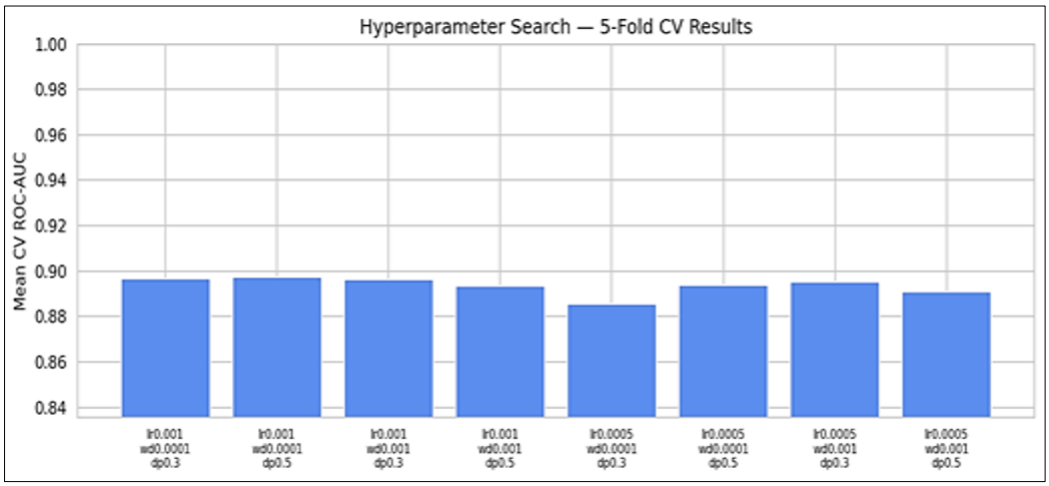


Fig 11: Mean five-fold cross-validation ROC-AUC for each hyperparameter configuration on the real data.

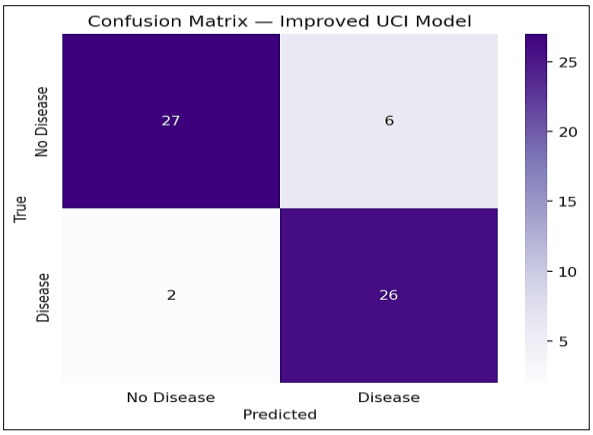


Fig 12: Confusion matrix on the real UCI test set, optimized model.

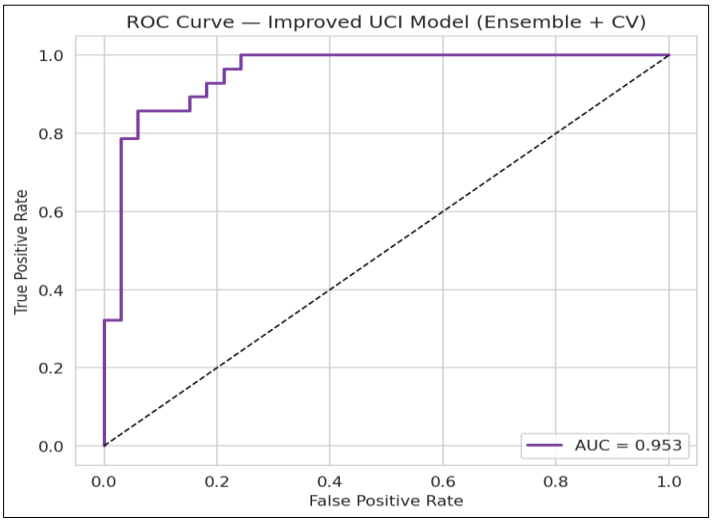


Fig 13: ROC curve on the real UCI test set, optimized model (AUC = 0.953).

8. Validation

A number of design features contribute to the reliability of the results reported; these are summarized.

8.1. Cross-validation and held-out testing

Model selection, using five-fold stratified cross-validation, gave an estimate of variance, which was necessary for the real data; a test set split before tuning gave an unbiased estimate. The standard deviations of the cross-validation were small, ranging from 0.02. The agreement between the cross-validation estimate for the selected configuration with the hold-out result, confirms that this agreement is consistent with no selection bias.

8.2. Class Imbalance Handling

Both are imbalanced sets of data. The penalty (Equation 6 and 7) was weighted against the class frequency. Results show the weighting to be effective with a balance of accuracy (precision) and sensitivity (recall) in each class, rather than a tendency to fall towards the majority class.

8.3. Guarding Against Information Leakage

Statistics and imputation values were only calculated on the training data and then used on the held-out data, and hyperparameters were only selected on the development partition. These precautions make sure that the test estimates are not over-inflated due to test information being used during the construction of the model.

8.4. Partial ablation of the optimization components

The cross validation search can be considered as a partial ablation over the strength of the regularization: weaker regularization resulted in lower mean ROC-AUC while stronger regularization resulted in the highest mean ROC-AUC, and regularization was found to be the only factor affecting the results. The improvement relative to the single split baseline is also an indication of variance reduction due to the cross-validated ensemble. A component-wise ablation is indicated as future work.

8.5. Stability of the performance estimates

The reported performance is not an artefact of a particular partition - the small standard deviations across cross-validation folds give a quantitative indication of this. For all 8 hyperparameter configurations examined, the mean fold ROC-AUC was kept within a narrow range and the fold-level standard deviation was kept near 2 hundredths, thus not influenced by noise but more stable to rank the configurations. The random seeds for the framework and numerical libraries were also fixed to ensure that the same results were obtained from the same model configuration when the model was rerun several times, which is essential for comparison between the baseline and optimized model. These properties suggest that the improvement in accuracy (8 points) over the real data represents a real effect of the optimization protocol and not just a sampling effect.

9. Comparative Analysis

Table 6 compares the optimized real-data model with some published studies of the UCI Cleveland benchmark. This accuracy of 0.869 is among the accuracy range of well validated models in the range from 87% to 92%. Results exceeding this range are often based on validation procedures that do not fully distinguish between model selection and testing and thus are not directly comparable with the rigorous testing procedure adopted here. The model is competitive to the existing classical and deep learning models within these limitations and has a compact architecture and transparent procedure. Gradient-boosted tree ensembles like XGBoost [38] and CatBoost [39] are 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, there are a number of different validation protocols, feature selection strategies and target encodings in the cited studies, and so the accuracies cited are indicative but not strictly commensurable. Secondly, the accuracy of the synthetic data should not be included in this table as if it was a result from real data, because that would be a mixing of two data regimes. The comparison is thus limited to the real-data model and the synthetic result is taken to be a statement about the synthetic task.

10. Discussion

10.1. Principal Findings

The analysis includes two findings. After transferring the architecture from synthetic to real data, accuracy was lost by about 8 points, while the ability to rank held up; this deficiency was corrected by regularization with the small sample. The second observation is the more informative one of the two measures of asymmetry. Accuracy dropped from 0.869 to 0.787 and ROC-AUC changed only from 0.940 to 0.920 when switching from the previous to the current data regime, meaning that the ordering of patients based on their predicted risk remained unchanged despite the change in data regime. An evaluation that would report only accuracy would have been a significant failure, an evaluation that would report only ROC-AUC alone would have reported for near-equivalence. Both of these readings are incomplete.

10.2. Why synthetic performance exceeded real performance

This was easier to classify than the real cohort because the synthetic corpus was made up of only a few texts. Several mechanisms are likely to contribute, and here only partly are separated from one another. Generators that preserve the marginal distributions accurately can still reduce the strength of higher-order relationships between variables, and if the relationships contain noise as well as signal in the actual patients, this reduction leads to more accurate and stable estimates of the class boundaries. The pattern that is observed here, with discrimination being transferred but not calibration, fits that account. Another possibility is reduced heterogeneity, a generated cohort has no measurement error, no missing entries, no rare presentations and these are found in the Cleveland records. Unlike real clinical data, there were no missing values nor any duplicate records in this synthetic dataset.

Work related to the problem supports the separation of statistical fidelity and downstream realism. A Bayesian network study of the same UCI family revealed that the size and internal consistency of the source cohort was consistent with the degree of benefit from synthetic augmentation, with the greatest benefit observed in the noisiest cohorts and the smallest benefits in the more complete cohorts, though these were not always statistically significant [26]. At large scales, it was found that synthetic records from a dual adversarial autoencoder enabled models that performed on par with models trained on real data, but inconsistencies in the models were revealed when running sex-stratified analysis that did not show up in aggregate utility metrics [27]. Both findings go in the same direction as the present study: that no matter how equal two groups are on a superficial measure, they are not equal on the measure that is clinically relevant.

10.3. The gap was remediable, not intrinsic

The second finding qualifies the first. The accuracy and ROC-AUC for real-data were finally 0.869 and 0.953 after early stopping, 5-fold cross validation, grid search and fold ensembling, equal to the synthetic accuracy and higher than its discrimination. The grid search picked out the best dropout rate, which directly controlled the mechanism, rather than by inference. The interaction of an expressive network and 242 training records was the limiter on the baseline, and not a characteristic of real clinical data per se. The implication in practice is that having small real cohorts does not necessarily limit the performance competitiveness as long as the additional variation due to the small cohort is explicitly controlled.

10.4. Implications for Model Development and Validation

These findings support a sequential workflow instead of an alternative. However, synthetic data can be used for prototyping architectures, debugging, and rehearsing analyses within privacy constraints, and cannot replace evaluation on real data. A study of colorectal cancer survival from the transfer learning perspective took another route: A model trained on synthetic data, and fine-tuned using local hospital data, outperformed a model trained only using local hospital data in the same settings, and the authors showed that synthetic data should augment rather than supplant real clinical models [28]. In that study, the performance for the direct zero-shot application was degraded in most configurations, the closest analogue to the untuned transfer reported here. The statements above are based on the architecture-controlled design. Comparing the results of the different studies does not allow for the separation of a difference in performance to data provenance. When all the three are kept constant and only the data is changed, the attribution is direct. The drawback of the design is its lack of breadth: it appeals to just one architecture family and two target groups and the size of the gap shown here should not be interpreted as a generally valid 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