

# ARO–FTAMNet: An Artificial Rabbits Optimization Guided Feature-Gated Multi-Scale Network for Heart Disease Prediction with External Validation

Rinaben Keshavlal Patel <sup>1</sup>, Prakash Arumugam <sup>1</sup>, Kaival Mehta <sup>2</sup>, Mainul Haque <sup>3,4,5,6,7</sup>

Please  
Click on  
Photo



## ABSTRACT

Heart disease remains the leading cause of death worldwide, and most machine-learning studies claiming to predict it are evaluated only on the same small training dataset. Such results say little about how a model will perform on new patients. This paper presents ARO (Artificial Rabbits Optimization)–FTAMNet (Future Tuned Adaptive Meta Network), a prediction framework trained on a mixed (hybrid) heart-disease corpus and tested on a separate patient set the model never encounters during training. The framework has two parts. The first is ARO, which simultaneously searches for a small, useful set of input attributes and good network settings, using a single fitness score that rewards ranking quality and balanced agreement while penalizing large feature sets. The second is FTAMNet, a compact feed-forward network with four working blocks: a dense projection, an adaptive feature gate that learns how much of each internal signal to retain, a residual learning block, and a three-branch multi-scale stage whose outputs are merged before a sigmoid decision layer. The decision threshold is not fixed at 0.5; it is tuned on validation data only and then frozen. On the external test set of 1,025 records, the framework achieves an accuracy of 0.7727, a precision of 0.7901, a recall of 0.7586, an F1-score of 0.7740, an AUROC (Area Under the Receiver Operating Characteristic Curve) of 0.8582, an AUPRC (Area Under The Precision-Recall Curve) of 0.8617, and a Matthews correlation coefficient of 0.5460. Three deep baselines built on the same data [Deep Multi-Layer Perceptron (DeepMLP), Residual Multi-Layer Perceptron (ResMLP), and Attention-based Multi-Layer Perceptron (AttMLP)] fall far behind, with MCC (Matthews Correlation Coefficient) values ranging from –0.005 to 0.078, indicating that plain deep networks do not transfer across cohorts. Removing ARO from the pipeline raises precision but cuts recall by about 15 percentage points, so the optimizer mainly buys sensitivity, which is what a screening tool needs. The study shows that a small, well-tuned network with an honest external test is more useful than a large model scored on its own data.

## Keywords

Heart Disease Prediction; Artificial Rabbits' Optimization; Feature Selection; Multi-Scale Neural Network; Adaptive Feature Gate; External Validation; Class Imbalance; Threshold Tuning.

## INTRODUCTION

Heart disease encompasses a group of disorders of the heart and blood vessels and is linked to about 17.9 million deaths every year, close to one-third of all deaths worldwide <sup>1</sup>. Global burden studies show that the number of cases continues to rise as populations age, and that a

1. Department of Computer Science and Engineering, Unitedworld Institute of Technology, Karnavati University, Gandhinagar, Gujarat, India.
2. Department of Smart Energy Learning Center (SELC), Dhirubhai Ambani University, Gandhinagar, Gujarat, India. Email: [kaivalhmehta@gmail.com](mailto:kaivalhmehta@gmail.com).
3. Department of Pharmacology and Therapeutics, Eastern Medical College and Hospital, Cumilla, Bangladesh. Email: [runurono@gmail.com](mailto:runurono@gmail.com).
4. Department of Research, Karnavati School of Dentistry, Karnavati University, Gandhi Nagar, Gujarat, India.
5. Scientific Committee, Global Alliance for Infections in Surgery, Macerata, Italy.
6. Department of Pharmacology and Therapeutics, National Defense University of Malaysia, Kuala Lumpur, Malaysia.
7. Public Health Foundation Bangladesh, Dhaka, Bangladesh.

## Correspondence

### 1. Rina Keshavlal Patel

Department of Computer Science and Engineering, Unitedworld Institute of Technology, Karnavati University, Gandhinagar, Gujarat, India. Email: [patelrina9686@gmail.com](mailto:patelrina9686@gmail.com)

### 2. Mainul Haque

Independent Researcher (Former Professor, National Defense University of Malaysia, Kuala Lumpur, Malaysia), Block: C, Road: 10, House: 266, Level 2. Khilgaon, Dhaka 1219, Bangladesh. Email: [runurono@gmail.com](mailto:runurono@gmail.com). Cell Phone & Whats App: +8801 898943255, +8801 703605918.

large share of early deaths occurs in low- and middle-income countries <sup>2</sup>. Because many risk factors can be addressed once identified, identifying high-risk people early is one of the most cost-effective ways to reduce this burden.

Machine learning has been applied to this task for more than two decades. Early work used logistic regression, decision trees, k-nearest neighbors, and support vector machines on small clinical datasets <sup>3,4</sup>. Later studies shifted to tree ensembles and gradient boosting, which still deliver the strongest results on tabular clinical data <sup>5,6</sup>, and more recent work has incorporated deep networks <sup>7,8</sup> and metaheuristic search for feature and parameter selection <sup>9,10</sup>. Reported accuracy in the literature is often above 95%.

These numbers are less impressive than they appear. Three problems keep recurring. First, most studies train and test on the same small benchmark, usually a few hundred records from the UC Irvine (UCI) Cleveland collection, and report only within-dataset scores. A score obtained this way measures internal consistency, not the ability to work on a new hospital or a new population. Second, the same few hundred records have been reused so often that the field has gradually adapted to them, which explains accuracies close to 100 percent. Third, many papers retain all input columns and use the default decision threshold of 0.5, even though the cost of missing a sick patient is far higher than the cost of a false alarm.

This work is built on a simple rule: a model is only as good as its score on data it has never seen. The proposed framework, ARO (Artificial Rabbits Optimization)–FTAMNet (Future Tuned Adaptive Meta Network), is trained on a hybrid heart-disease corpus and then tested once on a separate patient set that plays no role in training, validation, feature selection, or threshold tuning. Inside the pipeline, ARO <sup>11</sup> searches the space of input subsets and network settings jointly, guided by a fitness score that combines ranking quality with balanced agreement rather than plain accuracy. The selected inputs are then passed to FTAMNet. This small network combines a feature gate, a residual block, and three parallel branches of varying width, so that both fine and broad patterns are captured without a large parameter count.

The main contributions of the paper are as follows:

- A joint search procedure based on Artificial Rabbits

Optimization that selects the input attributes and the network settings in a single run, driven by a fitness score that rewards good ranking and balanced agreement while penalizing large subsets.

- FTAMNet, a compact network for tabular clinical data that uses an adaptive feature gate, a residual learning block, and a three-branch multi-scale stage, is described here with a complete set of equations.
- A decision threshold is tuned on validation data only and then frozen before the external test, so that no information from the test cohort reaches the model.
- An external test on 1,025 unseen records, reported with accuracy, precision, recall, specificity, F1-score, AUROC (Area Under the Receiver Operating Characteristic Curve), AUPRC (Area Under The Precision-Recall Curve), balanced accuracy, and the Matthews correlation coefficient (MCC), together with three deep baselines and an ablation isolating the effect of the optimizer.

The rest of the paper is organized as follows—Section 2 reviews related work. Section 3 describes the data and the framework. Section 4 presents the full mathematical model. Section 5 lists implementation details. Section 6 reports and discusses the results, and Section 7 concludes.

## RELATED WORK

### Classical and hybrid classifiers

Mohan et al. (2019) <sup>4</sup> combined a random forest with a linear model and reported approximately 88 percent accuracy on the UCI Cleveland dataset. Ali et al. (2019) <sup>12</sup> used stacked support vector machines for heart failure prediction, and Latha and Jeeva <sup>13</sup> used majority voting across several weak learners. All three gain a few points in accuracy over a single classifier, but they are evaluated on a few hundred near-balanced records, and none is tested on a second cohort. Ali et al. (2021) <sup>3</sup> compared a wide range of supervised methods on the same type of data and found that ensembles usually win, a result now well established.

### Deep networks for tabular clinical data

Dutta et al. (2020) <sup>7</sup> applied a convolutional network to coronary heart disease records and showed that deep models can read tabular data. Shrivastava et al. (2023) <sup>8</sup> combined a convolutional stage with a bidirectional

recurrent stage for the same task. These models learn their own features, but they require more data than most cardiac datasets provide and overfit quickly on small, balanced benchmarks. A broad comparison by Schwartz-Ziv and Armon (2022)<sup>14</sup> concluded that well-tuned gradient boosting still matches or outperforms deep networks on tabular problems of this size. The lesson here is that a deep model for this task should be small, regularized, and carefully tuned rather than deep for its own sake.

### Metaheuristic Search For Features And Settings

An exhaustive search is impossible because the number of possible input subsets grows to 2 to the power of the number of columns, so a nature-inspired search is used instead. The genetic algorithm<sup>15</sup> explores well but converges slowly; the gray wolf optimizer<sup>9</sup> converges quickly but tends to cluster around a single leader; particle swarm and its hybrids fall between the two<sup>10</sup>. ARO<sup>11</sup> is a more recent method that switches between

two behaviors — detour foraging for exploration and random hiding for exploitation — under the control of a single energy term that decreases as the search proceeds. This built-in switch is why it is adopted here. The main weakness of the existing search literature is the objective: most studies maximize plain accuracy, which is the wrong target whenever the classes are uneven, or the model will be applied to a cohort with a different disease rate.

### Position of this work

Table 1 places the present work alongside representative recent studies. The pattern is clear: strong results in one column are often offset by an empty cell in another. Studies with high accuracy are tested on a single small cohort; studies with clever search optimize accuracy alone; studies with deep models rarely report a threshold-free metric. The framework proposed here is deliberately assembled to fill those cells at once.

**Table 1:** Recent heart-disease prediction studies compared with the present work.

| Study (year)                           | Method                                                                                    | Data used                                     | Reported result                             | Main gap                                             |
|----------------------------------------|-------------------------------------------------------------------------------------------|-----------------------------------------------|---------------------------------------------|------------------------------------------------------|
| Mohan et al. (2019) <sup>4</sup>       | Random forest + linear model                                                              | UC Irvine (UCI) Cleveland (~300)              | ≈88% accuracy                               | Small balanced data; accuracy only; no external test |
| Ali et al. (2019) <sup>12</sup>        | Stacked SVM (Support Vector Machines) expert system                                       | Heart failure (small)                         | ≈92% accuracy                               | No imbalance handling; single cohort                 |
| Latha and Jeeva (2019) <sup>13</sup>   | Majority-vote ensemble                                                                    | UCI Cleveland                                 | +7% over single model                       | Tiny dataset; no threshold-free metric               |
| Dutta et al. (2020) <sup>7</sup>       | Convolutional network                                                                     | Coronary records                              | AUC (Area under curve) above shallow models | Opaque; no feature selection                         |
| Ali et al. (2021) <sup>3</sup>         | Comparison of classical methods                                                           | UCI-type records                              | Ensembles best                              | Benchmark only; no transfer test                     |
| Rajammal et al. (2022) <sup>10</sup>   | Binary GWO (Gray Wolf Optimizer) + mutation                                               | Biomedical benchmarks                         | Compact subsets                             | Accuracy objective; not cardiac                      |
| Shrivastava et al. (2023) <sup>8</sup> | CNN (Convolutional Neural Network) + BiLSTM (Bidirectional Long Short-Term Memory) hybrid | Heart-disease records                         | High accuracy                               | Heavy model; single cohort                           |
| Wang et al. (2022) <sup>11</sup>       | ARO                                                                                       | Engineering benchmarks                        | Strong convergence                          | Not applied to clinical screening                    |
| This work                              | ARO search + FTAMNet + tuned threshold                                                    | Hybrid corpus (2,283) trained, 1,025 external | AUROC 0.8582, MCC 0.5460 on unseen data     | Addresses the gaps listed above                      |

## MATERIALS AND METHODS

### Data

Two separate collections are used and kept strictly apart. The first is a hybrid training corpus of 2,283 patient records compiled from public heart-disease sources that share a common set of clinical fields, including age, sex, chest-pain type, resting blood pressure, serum cholesterol, fasting blood sugar, resting electrocardiogram result, maximum heart rate, exercise-induced angina, ST depression, slope of the ST segment, number of major vessels, and thalassemia status.

The target is a binary label indicating the presence or absence of heart disease. This corpus is split once, with stratification, into 1,826 training records (80%) and 457 validation records (20%).

The second collection is an external test set of 1,025 records from an independent source. It contains 526 positive and 499 negative cases, yielding a positive rate of 51.3%. This set is used only once, at the very end, to produce all numbers reported in Section 6. It is not used for cleaning decisions, feature selection, network tuning, or threshold selection. Table 2 summarizes both collections.

**Table 2:** Data used in the study. The external set is never touched during model building.

| Collection                      | Records | Role          | Positive rate         | Used for                                        |
|---------------------------------|---------|---------------|-----------------------|-------------------------------------------------|
| Hybrid corpus – training part   | 1,826   | Training      | Balanced after merge  | Network weights, ARO fitness                    |
| Hybrid corpus – validation part | 457     | Validation    | Same as training part | Fitness scoring, early checks, threshold tuning |
| External heart dataset          | 1,025   | External test | 51.3% (526 / 1,025)   | Final reporting only                            |

### Data Preparation

Preparation is kept simple so it can be repeated on a new cohort. Duplicate rows are removed. Missing entries in continuous fields are filled with the column median, and missing entries in categorical fields with the column mode, both computed only on the training set; the resulting values are then applied to the validation and external sets. Categorical fields with a natural order, such as chest-pain grade, are given integer codes that preserve that order, while fields without an order are one-hot coded. Continuous fields are scaled to zero mean and unit variance using statistics from the training set alone, preventing information from the test cohort from leaking into the model. Class weights are used in the loss function instead of synthetic oversampling, because generating artificial patients can move the decision surface into regions where no real patients exist.

### Framework Overview

Figure 1 shows the complete pipeline. The hybrid corpus enters the preprocessing block. ARO then searches for a good feature subset and a good set of network settings at the same time; each candidate is scored by training a light version of FTAMNet and measuring its fitness

on the validation part. The best candidate found is used to build the final FTAMNet, which is trained on the selected inputs. The decision threshold is tuned on the validation part, and only then is the frozen model applied to the external dataset. Performance is reported with six standard measures.

### Mathematical Model

This section states the problem, the search algorithm, the network, and the scoring rules in full. Every symbol used in an equation is explained directly below it.

### Problem statement

Let the training data be a set of labeled patient records:

$$D = \{ (x_i, y_i) : i = 1, 2, \dots, N \}, \quad x_i \in \mathbb{R}^d, \quad y_i \in \{0, 1\} \quad (1)$$

$D$  is the labeled dataset used for learning.

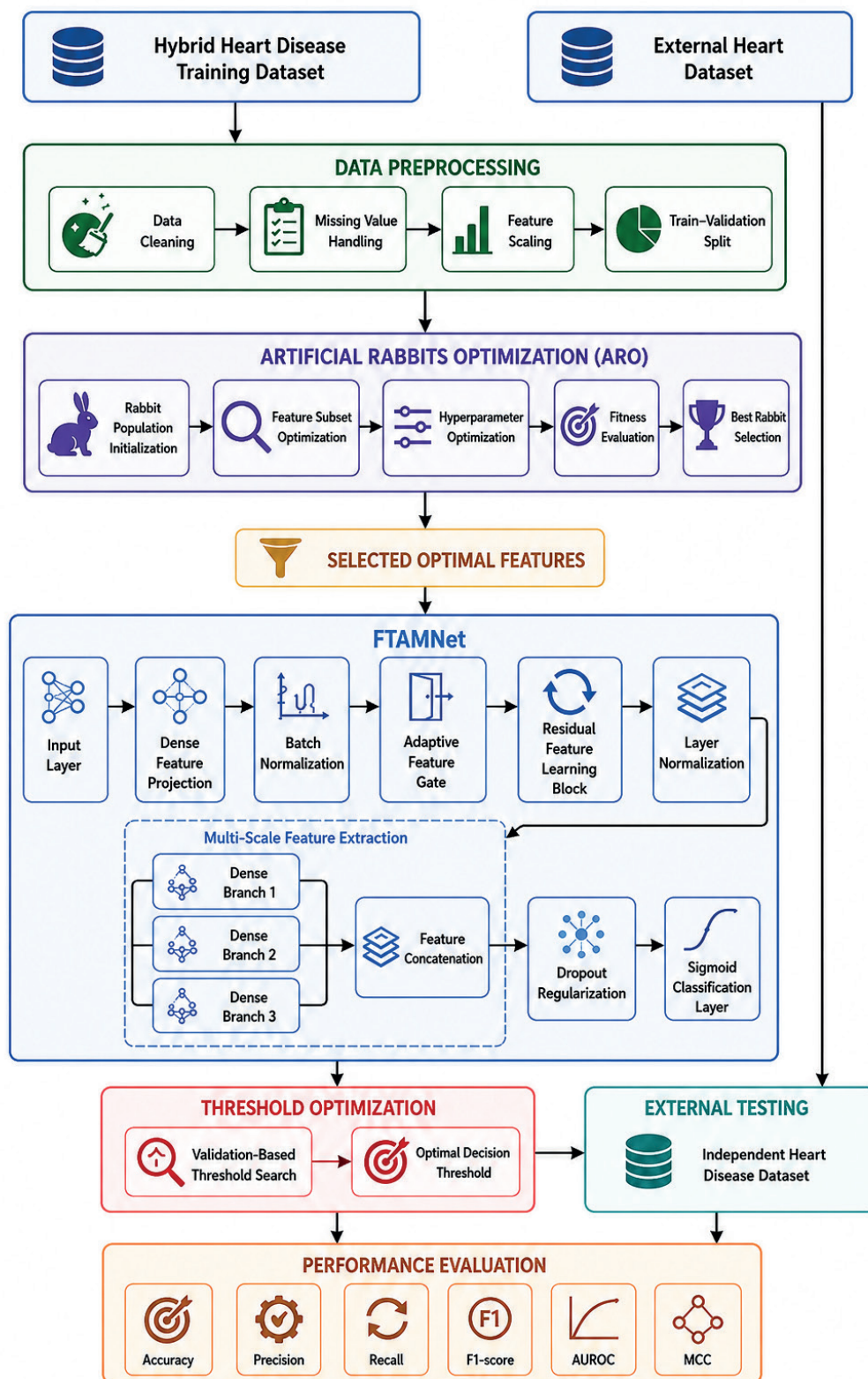
$x_i$  is the vector of clinical measurements of patient  $i$ .

$y_i$  is the label of patient  $i$ , where 1 means heart disease is present, and 0 means it is absent.

$N$  is the number of records in the set.

$d$  is the number of input attributes before selection.

The task is to learn a scoring function together with a



**Figure 1.** End-to-end structure of the proposed ARO-FTAMNet framework, from data preparation and rabbit-based search through the FTAMNet blocks, threshold tuning, and external testing.

**Figure Credit:** Rinaben Keshavlal Patel.

cut-off value:

$$f\theta: \mathbb{R}^d \rightarrow [0, 1], \quad \hat{y}_i = 1 \text{ if } f\theta(x_i) \geq \tau, \text{ otherwise } \hat{y}_i = 0(2)$$

$f\theta$  is the network that turns a record into an estimated probability of disease.

$\theta$  are all trainable weights and biases of that network.

$\hat{y}_i$  is the predicted class for patient  $i$ .

$\tau$  is the decision threshold, which is tuned rather than fixed at 0.5.

Feature selection is written as a mask applied to the input:

$$x_i^s = x_i \square s, \quad s \in \{0, 1\}^d, \quad |s| = \sum_j s_j(3)$$

$s$  the binary mask; entry  $j$  equals 1 when attribute  $j$  is kept and 0 when it is dropped.

$\square$  element-by-element multiplication.

$|s|$  the number of attributes that survive the mask.

$x_i^s$  the reduced record that is fed to the network.

### Artificial Rabbits Optimization

ARO copies two habits of wild rabbits. When a rabbit is fresh, it feeds far from its own burrow, spreading the search over the whole space; when it is tired, it hides in one of its own burrows, refining the area already found. A single energy term determines which habit is used at each step<sup>11</sup>.

Each candidate solution is one rabbit, and its position holds both the feature mask and the network settings:

$$z_i = [u_{i,1}, \dots, u_{i,d}, v_{i,1}, \dots, v_{i,q}](4)$$

$z_i$  is the position of rabbit  $i$  in the search space.

$u_{i,j}$  a real value that will be turned into the mask bit of attribute  $j$ .

$v_{i,k}$  a real value in  $[0, 1]$  that encodes the  $k$ -th network setting.

$q$  the number of network settings under search (learning rate, dropout rate, projection width, branch width, and batch size).

The first  $d$  values are turned into mask bits with a sigmoid transfer function:

$$s_j = 1 \text{ if } 1 / (1 + e^{-u_j}) \geq 0.5, \text{ otherwise } s_j = 0(5)$$

$s_j$  is the mask bit of attribute  $j$ .

$u_j$  is the real value produced by the search for

attribute  $j$ .

$e$  is the base of the natural logarithm.

The remaining values are stretched onto the range of each setting:

$$h_k = h_k^{\min} + (h_k^{\max} - h_k^{\min}) \cdot v_k(6)$$

$h_k$  the decoded value of network setting  $k$ .

$h_k^{\min}, h_k^{\max}$  the lowest and highest values allowed for that setting (Table 3).

$v_k$  the search value in  $[0, 1]$  taken from the rabbit position.

The energy term controls the switch between the two behaviors:

$$A(t) = 4 \cdot (1 - t/T) \cdot \ln(1/r)(7)$$

$A(t)$  is the energy of the population at iteration  $t$ ; exploration is used while  $A(t) > 1$  and exploitation while  $A(t) \leq 1$ .

$t$  the current iteration number.

$T$  the total number of iterations allowed.

$r$  a random number drawn uniformly from  $(0, 1)$ .

$\ln$  the natural logarithm.

When the energy is high, a rabbit moves towards a randomly chosen companion; this is detouring foraging:

$$w_i(t+1) = z_j(t) + R \cdot (z_i(t) - z_j(t)) + \text{round}(0.5 \cdot (0.05 + r_1)) \cdot n_1, \quad j \neq i(8)$$

$w_i(t+1)$  the trial position produced for rabbit  $i$ .

$z_i(t), z_j(t)$  the current positions of rabbit  $i$  and of a randomly picked rabbit  $j$ .

$R$  the running operator defined in the next equation.

$r_1$  a uniform random number in  $[0, 1]$ .

$n_1$  a random value drawn from the standard normal distribution, which adds a small jump.

$\text{round}(\cdot)$  rounding to the nearest whole number.

The running operator gives the search its zig-zag movement:

$$R = L \cdot c, \quad L = (e - e^{-(t-1)/T}) \cdot \sin(2\pi \text{ or } 2)(9)$$

$L$  the running length, which shrinks as the iterations go on.

$c$  a mapping vector, defined next, that lets only part of the coordinates move.

$r_2$  a uniform random number in  $[0, 1]$ .

$\sin$  the sine function, which makes the step change direction.

$$c_k = 1 \text{ if } k \in G(l), \text{ otherwise } c_k = 0, \quad l = \lfloor r_3 \cdot (d + q) \rfloor (10)$$

$c_k$  entry  $k$  of the mapping vector.

$G(l)$  a set of  $l$  positions taken from a random permutation of the coordinate indices.

$l$  the number of coordinates allowed to change in this step.

$d + q$  the length of the position vector, that is, the feature genes plus the setting genes.

$\lfloor \cdot \rfloor$  rounding up to the next whole number.

$r_3$  a uniform random number in  $[0, 1]$ .

When the energy falls, the rabbit digs burrows around its own position and hides in one of them; this is random hiding:

$$b_{i,j}(t) = z_i(t) + H \cdot g_j \cdot z_i(t), \quad H = ((T - t + 1) / T) \cdot r_4 (11)$$

$b_{i,j}(t)$  the  $j$ -th burrow of rabbit  $i$ .

$H$  the hiding parameter, which becomes smaller as the search advances so that burrows stay close to the rabbit.

$g_j$  a vector whose entry  $j$  is 1 and all other entries are 0, so only one coordinate is shifted.

$r_4$  a uniform random number in  $[0, 1]$ .

$$w_i(t+1) = z_i(t) + R \cdot (r_5 \cdot b_{i,r}(t) - z_i(t)) (12)$$

$b_{i,r}(t)$  one burrow chosen at random from those built by rabbit  $i$ .

$r_5$  a uniform random number in  $[0, 1]$ .

$R$  the running operator already defined.

A trial position is accepted only when it is better than the current one:

$$z_i(t+1) = w_i(t+1) \text{ if } F(w_i(t+1)) < F(z_i(t)), \text{ otherwise } z_i(t+1) = z_i(t) (13)$$

$F(\cdot)$  is the fitness function defined below, which is minimized.

$z_i(t+1)$  is the position carried into the next iteration.

### Fitness function

The fitness must reflect what a screening tool is asked to do: rank patients well, agree with the truth on both classes, and stay small. It is therefore written as a cost that mixes three terms:

$$F(z) = w_1 \cdot (1 - AUROC_{val}) + w_2 \cdot (1 - MCC_{val}) + \lambda \cdot (|s| / d) (14)$$

$F(z)$  is the cost of the candidate rabbit  $z$ ; lower is better.

$AUROC_{val}$  is the area under the ROC curve of the candidate model on the validation set.

$MCC_{val}$  is the Matthews correlation coefficient of the candidate model on the validation set.

$w_1, w_2$  weights that set the balance between ranking quality and balanced agreement.

$\lambda$  a small penalty weight that discourages large feature subsets.

$|s| / d$  the share of attributes kept by the candidate.

Plain accuracy is intentionally left out. Accuracy can be pushed up by always predicting the larger class, but it changes whenever the disease rate of the cohort shifts, so it is a poor guide for a model meant to generalize to new populations.

### The FTAMNet network

FTAMNet takes the selected inputs and passes them through five stages. The activation used inside the network is the rectified linear unit:

$$\phi(a) = \max(0, a) (15)$$

$\phi(\cdot)$  is the activation function applied element by element.

$a$  the value entering the activation.

Stage 1 — dense feature projection. The reduced record is lifted into a wider space so that the later blocks have room to work:

$$h^{(0)} = \phi(W_p x^s + b_p) (16)$$

$h^{(0)}$  is the projected representation.

$W_p$  is the weight matrix of the projection layer, of size  $m \times |s|$ .

$b_p$  is the bias vector of the projection layer.

$m$  is the projection width, one of the settings chosen by ARO.

Stage 2 — batch normalization. The projected values are standardized so that training stays stable:

$$h^{(1)} = \gamma \cdot (h^{(0)} - \mu_B) / \sqrt{(\sigma_B^2 + \varepsilon)} + \beta (17)$$

$\mu_B, \sigma_B^2$  are the mean and variance of the current mini-batch, taken per unit.

$\gamma, \beta$  are trainable scale and shift parameters.

$\varepsilon$  is a very small constant that prevents division by zero.

Stage 3 — adaptive feature gate. The network learns how much of each internal signal deserves to pass. This is the attention step, and it is deliberately cheap:

$$g = \sigma(W_g h^{(1)} + b_g), \sigma(a) = 1 / (1 + e^{-a}) \quad (18)$$

$g$  the gate vector; every entry lies between 0 and 1 and acts as an importance weight.

$W_g, b_g$  the weights and bias of the gate layer.

$\sigma(\cdot)$  the logistic function, which keeps the gate inside  $[0, 1]$ .

$$h^{(2)} = g \square h^{(1)} \quad (19)$$

$h^{(2)}$  the gated representation, in which weak signals are damped and useful ones are kept.

$\square$  element-by-element multiplication.

Stage 4 — residual feature learning block followed by layer normalization. The block learns a correction and adds it back to its own input, which keeps the gradient path short:

$$h^{(3)} = h^{(2)} + \varphi(W_2 \varphi(W_1 h^{(2)} + b_1) + b_2) \quad (20)$$

$W_1, b_1$  are the weights and bias of the first layer inside the block.

$W_2, b_2$  are the weights and bias of the second layer inside the block.

$h^{(3)}$  is the output of the residual block.

$$h^{(4)} = \gamma_L \cdot (h^{(3)} - \mu_h) / \sqrt{(\sigma_h^2 + \varepsilon)} + \beta_L \quad (21)$$

$\mu_h, \sigma_h^2$  the mean and variance taken across the units of a single sample;

$\gamma_L, \beta_L$  trainable scale and shift parameters of the layer-normalization step;

$h^{(4)}$  the normalized representation passed to the multi-scale stage.

Stage 5 — multi-scale feature extraction. Three parallel branches of different width look at the same representation at different levels of detail, and their outputs are joined:

$$m^{(k)} = \varphi(W^k h^{(4)} + b^k), \quad k = 1, 2, 3 \quad (22)$$

$m^{(k)}$  the output of branch  $k$ ;

$W^k, b^k$  the weights and bias of branch  $k$ ;

$k$  the branch index; the three branches use decreasing widths so that branch 1 captures broad patterns and branch 3 captures narrow ones.

$$u = [m^{(1)}; m^{(2)}; m^{(3)}] \quad (23)$$

$u$  the joined feature vector formed by placing the three branch outputs one after another;

$[\cdot; \cdot]$  concatenation along the feature axis.

Dropout is then applied so that the network does not lean on any single unit:

$$\tilde{u} = (u \square m) / (1 - p_d), \quad m_j \sim \text{Bernoulli}(1 - p_d) \quad (24)$$

$\tilde{u}$  the vector after dropout;

$m_j$  a random switch that is 1 with probability  $1 - p_d$  and 0 otherwise;

$p_d$  the dropout rate, one of the settings chosen by ARO;

$1 / (1 - p_d)$  the scaling factor that keeps the expected value unchanged.

The final layer produces the estimated probability of disease:

$$\hat{p} = \sigma(w_o^T \tilde{u} + b_o) \quad (25)$$

$\hat{p}$  the predicted probability that the patient has heart disease;

$w_o, b_o$  the weight vector and bias of the output unit;

$\sigma(\cdot)$  the logistic function defined earlier.

### Training objective and threshold tuning

The network is trained with a class-weighted cross-entropy loss so that the rarer class is not ignored:

$$\mathcal{L} = - (1/N) \sum_i [\alpha_+ y_i \log \hat{p}_i + \alpha_- (1 - y_i) \log(1 - \hat{p}_i)] \quad (26)$$

$\mathcal{L}$  the loss to be minimized;

$N$  the number of records in the training part;

$y_i, \hat{p}_i$  the true label and the predicted probability of record  $i$ ;

$\alpha_+, \alpha_-$  the weights of the positive and negative class, defined next.

$$\alpha_+ = N / (2 N_+), \quad \alpha_- = N / (2 N_-) \quad (27)$$

$N_+, N_-$  are the numbers of positive and negative records in the training part; the rarer class receives the larger weight.

The weights are updated with the Adam rule:

$$\theta_{t+1} = \theta_t - \eta \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \varepsilon) \quad (28)$$

$\theta_t$  the parameter values at step  $t$ ;

$\eta$  the learning rate found by ARO;

$\hat{m}_t, \hat{v}_t$  the bias-corrected running averages of the gradient and of the squared gradient;

$\varepsilon$  a small constant for numerical safety.

Finally, the decision threshold is chosen on the validation part only:

$$\tau^* = \arg \max_{\{\tau \in (0,1)\}} MCC_{val}(\tau) \quad (29)$$

$\tau^*$  is the operating threshold that is then frozen and applied unchanged to the external set;

$MCC_{val}(\tau)$  is the Matthews correlation coefficient obtained on the validation part when  $\tau$  is used as the cut-off.

### Scoring rules

All results are reported from the four counts of the confusion matrix: true positives TP, false positives FP, true negatives TN and false negatives FN.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (30)$$

$$Precision = TP / (TP + FP), \quad Recall = TP / (TP + FN) \quad (31)$$

$$Specificity = TN / (TN + FP), \quad Balanced accuracy = (Recall + Specificity) / 2 \quad (32)$$

TP sick patients correctly flagged;

FP healthy patients wrongly flagged;

TN healthy patients correctly cleared;

FN sick patients missed by the model, the costliest error in screening.

$$F1 = 2 \cdot (Precision \cdot Recall) / (Precision + Recall) \quad (33)$$

F1 is the harmonic mean of precision and recall; it stays low unless both are reasonable.

$$MCC = (TP \cdot TN - FP \cdot FN) / \sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)} \quad (34)$$

MCC is a single balanced score that uses all four counts; it lies in  $[-1, 1]$ , where 1 is perfect agreement, 0 is guessing, and -1 is total disagreement<sup>16</sup>.

$$AUROC = \int_0^1 TPR(FPR) d(FPR) \quad (35)$$

AUROC is the area under the curve of true-positive rate against false-positive rate; it measures ranking quality without fixing a threshold;

TPR and FPR are the true-positive rate (recall) and

the false-positive rate, both computed as the threshold is swept from 1 to 0.

$$AUPRC = \int_0^1 Precision(Recall) d(Recall) \quad (36)$$

AUPRC is the area under the precision–recall curve, which focuses on the positive class and is the more honest of the two areas when the classes are uneven.

$$Brier = (1/N) \sum_i (\hat{p}_i - y_i)^2 \quad (37)$$

Brier is the mean squared gap between the predicted probability and the true label; it measures whether the probabilities themselves can be trusted, not just their order.

### Implementation Details

#### Search Settings

Table 3 lists the search space given to ARO and the fitness weights. The ranges are wide enough to allow a genuinely different network to appear but narrow enough that every candidate can be trained for a short number of epochs during scoring, which keeps the search affordable.

**Table 3:** Search space and fitness settings supplied to Artificial Rabbits Optimization.

| Item                           | Range or rule                            | Purpose                                           |
|--------------------------------|------------------------------------------|---------------------------------------------------|
| Population size                | 20 rabbits                               | Number of candidate solutions kept alive          |
| Iterations (T)                 | 30                                       | Length of the search                              |
| Feature genes                  | d binary bits after sigmoid transfer     | Choose which attributes are kept                  |
| Learning rate ( $\eta$ )       | $1 \times 10^{-4}$ to $1 \times 10^{-2}$ | Step size of the Adam update                      |
| Dropout rate ( $p_d$ )         | 0.10 to 0.50                             | Strength of regularization                        |
| Projection width (m)           | 64 to 256 units                          | Width of the first dense layer                    |
| Branch widths                  | 16 to 64 units per branch                | Detail level of the three branches                |
| Batch size                     | 32, 64 or 128                            | Records per gradient step                         |
| Fitness weights ( $w_1, w_2$ ) | 0.6 and 0.3                              | Balance of ranking quality and balanced agreement |
| Sparsity weight ( $\lambda$ )  | 0.1                                      | Penalty on the share of attributes kept           |

## Final Configuration

The best rabbit found by the search defines both the input subset and the network. The settings carried into the final training run are listed in Table 4. The learning rate of  $3.35 \times 10^{-3}$  is the value returned by the optimizer; it is higher than the value typically chosen by hand for a network of this size, and it works because the batch-normalization and layer-normalization steps keep the activations under control. During training, this rate was halved once, at epoch 16, when the validation loss stopped improving.

**Table 4:** Final Configuration Of The ARO–Ftmnet Model Used For All Reported Results.

| Component             | Setting                                                             | Note                                                                           |
|-----------------------|---------------------------------------------------------------------|--------------------------------------------------------------------------------|
| Input                 | ARO-selected subset of the clinical attributes                      | The retained attributes are listed in the optimizer log released with the code |
| Dense projection      | Single layer, ReLU                                                  | Widens the reduced record                                                      |
| Batch normalization   | Applied after projection                                            | Keeps activations stable                                                       |
| Adaptive feature gate | Sigmoid gate, element-wise product                                  | Learns how much of each signal to keep                                         |
| Residual block        | Two layers with a skip connection                                   | Short gradient path                                                            |
| Layer normalizations  | Applied after the residual block                                    | Per-sample stabilization                                                       |
| Multi-scale stage     | Three parallel dense branches, joined                               | Broad and narrow patterns together                                             |
| Dropout               | Applied before the output unit                                      | Prevents reliance on single units                                              |
| Output                | One unit with sigmoid activation                                    | Probability of heart disease                                                   |
| Optimizer             | Adam                                                                | Adaptive step sizes                                                            |
| Initial learning rate | $3.35 \times 10^{-3}$ (halved to $1.68 \times 10^{-3}$ at epoch 16) | Value returned by ARO                                                          |
| Loss                  | Class-weighted binary cross-entropy                                 | Protects the smaller class                                                     |
| Epochs                | 20                                                                  | Best validation checkpoint retained                                            |
| Decision threshold    | $\tau^* = 0.27$ , tuned on validation only                          | Frozen before the external test                                                |

## Tools and Hardware

The framework was written in Python 3.10. The network was built with TensorFlow and Keras; data handling was done with pandas and NumPy; and scoring was done with scikit-learn. The ARO search was implemented in-house from the published description of the algorithm 11. Figures were produced with Matplotlib and saved at 600 dpi. All experiments were run on a workstation with a multi-core x86-64 processor, 32 GB of memory, and a GPU with 8 GB of video memory. Because the network is small, the full search and final training together complete within a normal working session on this machine, and the trained model can be applied to a new cohort in a fraction of a second per record.

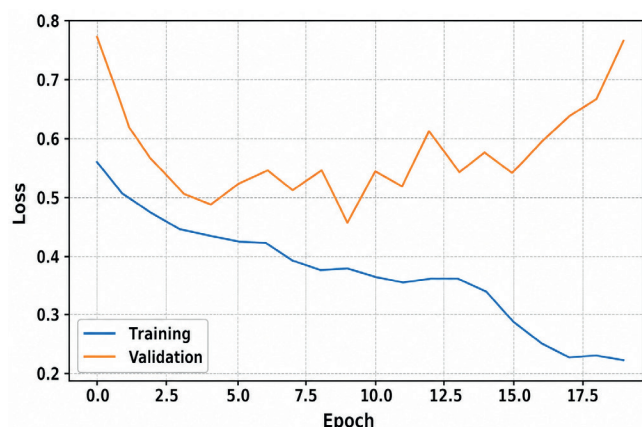
## RESULTS

### Training Behavior

Figure 2 shows the training and validation loss over the twenty epochs of the final run. The two curves coincide for the first ten epochs, which shows that the network is learning structure that holds beyond the training records. The lowest validation loss, 0.4620, is reached at epoch 10, where the validation AUROC is 0.8742, and the validation accuracy is 0.7877. After epoch 15, the training loss continues to decline, from 0.352 to 0.233, while the validation loss rises to 0.764. That gap is ordinary overfitting on a corpus of this size, and it is the reason the checkpoint with the best validation score, rather than the last checkpoint, was kept for testing. The single drop in the learning rate at epoch 16 is visible as a sharp fall in the training curve.

### Performance on the External Dataset

Table 5 reports the results of applying the frozen model to the 1,025 unseen records. The model achieves an accuracy of 0.7727 and, more importantly, maintains high precision and recall of 0.7901 and 0.7586, respectively. Specificity is 0.7876, so the balanced accuracy of 0.7731 is nearly the same as the plain accuracy; the model is not biased toward one class. The AUROC of 0.8582 and the AUPRC of 0.8617 indicate that the ranking is sound at every threshold, not just the chosen one, and the MCC of 0.5460 confirms moderate but real agreement with the truth on both classes. For a model that has never seen a single record from this cohort, these numbers fairly reflect what the framework would deliver in practice.



**Figure 2:** Training and validation losses of FTAMNet over the 20 training epochs.

**Figure Credit:** Rinaben Keshavlal Patel.

**Table 5:** ARO-FTAMNet on the external test set of 1,025 unseen records, at the frozen threshold  $\tau^* = 0.27$ .

| Measure              | Value  | Reading                                               |
|----------------------|--------|-------------------------------------------------------|
| Accuracy             | 0.7727 | Just over three quarters of records classed correctly |
| Precision            | 0.7901 | About four in five alerts are genuine                 |
| Recall (sensitivity) | 0.7586 | About three in four sick patients are found           |
| Specificity          | 0.7876 | About four in five healthy patients are cleared       |
| F1-score             | 0.7740 | Precision and recall are well balanced                |
| Balanced accuracy    | 0.7731 | No hidden bias towards either class                   |
| AUROC                | 0.8582 | Good ranking at any threshold                         |
| AUPRC                | 0.8617 | Good ranking of the positive class                    |
| MCC                  | 0.5460 | Moderate agreement using all four counts              |
| Brier score          | 0.1958 | Probabilities are usable but not sharp                |

### Comparison with Deep Baselines

Three deep baselines were built on the same data and scored the same way: DeepMLP (Deep Multi-Layer

Perceptron), a plain stack of dense layers; ResMLP (Residual Multi-Layer Perceptron), the same stack with skip connections; and AttMLP (Attention-based Multilayer Perceptron), a dense stack with an attention layer. Table 6 shows what happens when all four models are evaluated on the external cohort. The three baselines collapse. Their accuracies are near the level of coin tossing, their recall values are below 0.14, and their MCC values range from  $-0.005$  to  $0.078$ , indicating they carry almost no information about the new patients. DeepMLP is the worst case: it is not merely weak but slightly worse than random, so the pattern it learned on the training corpus is actively wrong on the new one. ARO-FTAMNet, by contrast, maintains an MCC of  $0.5460$  and an AUROC of  $0.8582$ .

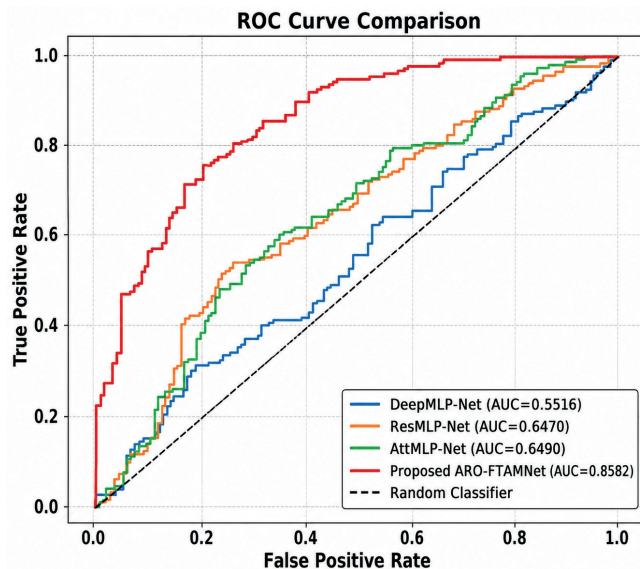
The size of the gap is the main finding of this study. All four models were trained on the same records with the same preprocessing, so the data cannot explain the difference. It stems from three design choices working together: a reduced input set that removes attributes whose meaning shifts across cohorts, a fitness score that rewards balanced agreement rather than accuracy, and a task-specific threshold rather than a fixed  $0.5$ . Any one of these, on its own, would help; together, they make the difference between a model that transfers and one that does not.

**Table 6:** Performance of all four models on the external test set. Each model uses its own tuned threshold; AUROC does not depend on the threshold.

| Model       | Accuracy | Precision | Recall | F1     | AUROC  | MCC       |
|-------------|----------|-----------|--------|--------|--------|-----------|
| DeepMLP     | 0.4868   | 0.5000    | 0.0342 | 0.0641 | 0.5516 | $-0.0050$ |
| ResMLP      | 0.5015   | 0.6056    | 0.0817 | 0.1441 | 0.6470 | 0.0505    |
| AttMLP      | 0.5141   | 0.6228    | 0.1350 | 0.2219 | 0.6490 | 0.0776    |
| ARO-FTAMNet | 0.7727   | 0.7901    | 0.7586 | 0.7740 | 0.8582 | 0.5460    |

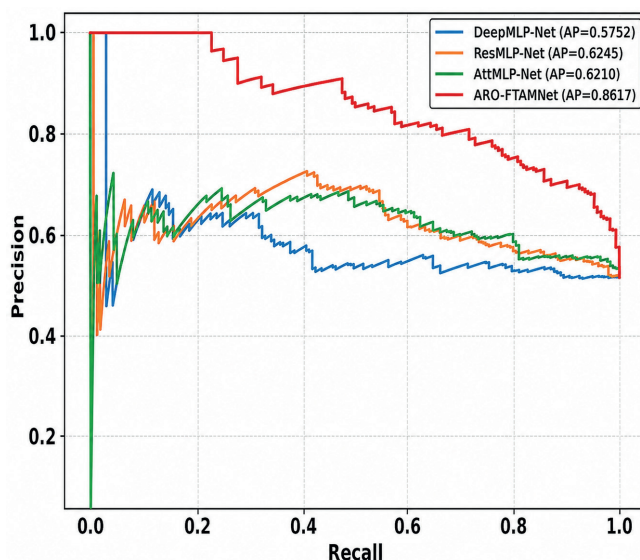
Figure 3 shows the ROC (Receiver Operating Characteristic) curves for the four models, and Figure 4 shows the precision-recall curves. The ROC curve for ARO-FTAMNet sits well above the others across the entire range. At the same time, the three baselines

hug the diagonal, which is characteristic of a model with no transferable knowledge. The precision–recall view is harsher and therefore more informative: the baselines lose precision as soon as recall exceeds a small threshold, whereas the proposed model maintains precision above 0.75 until recall exceeds 0.80.



**Figure 3:** ROC curves of the four models on the external test set.

**Figure Credit:** Rinaben Keshavlal Patel.

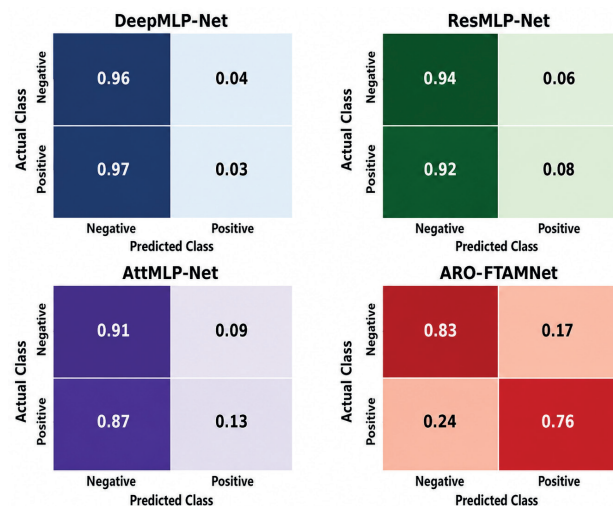


**Figure 4:** Precision–recall curves of the four models on the external test set.

**Figure Credit:** Rinaben Keshavlal Patel.

## Error analysis

Figure 5 shows the confusion matrices of all four models side by side. For ARO–FTAMNet, the counts are 393 true negatives, 106 false positives, 127 false negatives, and 399 true positives. In screening terms, the 127 missed cases are the expensive errors, and the 106 false alarms are the cheap ones, since a false alarm leads only to a further check, while a missed case leads to nothing at all. The two error types are similar in size here, which is the direct effect of tuning the threshold on MCC rather than using 0.5. At the default threshold, the same model produces 265 missed cases instead of 127, more than double, in exchange for only 64 fewer false alarms. That single comparison shows how much of the useful behavior of a screening model lives in the threshold rather than in the weights.



**Figure 5:** Confusion matrices of the four models on the external test set.

**Figure Credit:** Rinaben Keshavlal Patel.

## Effect of the Optimizer

To separate the contribution of ARO from that of the network, FTAMNet was also trained using all attributes and hand-set values in place of the searched ones. Table 7 compares the two versions on the same external records. The plain version has higher precision (0.8726 vs. 0.7901) and higher AUROC (0.9033 vs. 0.8582). The searched version has a wider margin: higher recall (0.7586 versus 0.6122) and higher F1-score, accuracy, and MCC.

This trade is worth stating plainly rather than hiding. ARO does not make the model a better ranker on this

cohort; the drop in AUROC shows the opposite. What it does is move the model to an operating point that identifies far more sick patients, raising recall by about 15 percentage points at the cost of about 8 percentage points of precision; in a screening setting, that is the right direction, because a missed case has a much higher cost than a repeat test. The F1-score and the MCC, which are the two measures that weigh both kinds of error, both improve, so the exchange is favorable by the measures that matter. It should also be noted that a single AUROC difference of 0.045 in one cohort is not strong evidence for ranking in general, and a study across several cohorts would be needed to settle that point.

**Table 7:** Ablation on the external test set: FTAMNet with hand-set values and all attributes against the full ARO-FTAMNet.

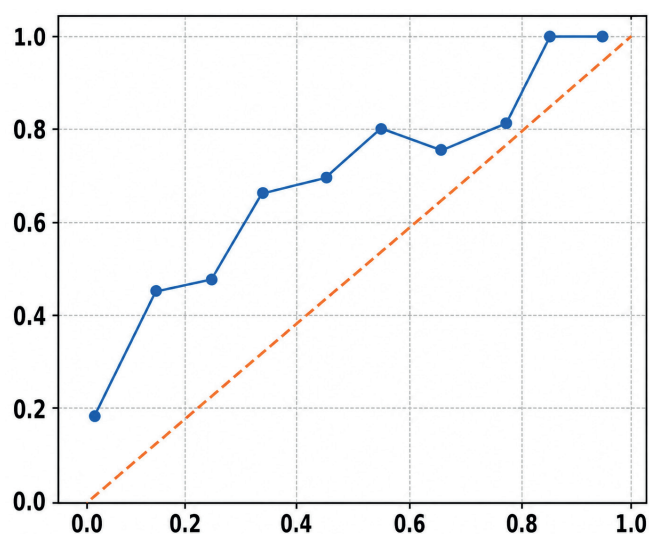
| Measure   | FTAMNet without ARO | ARO-FTAMNet | Change  |
|-----------|---------------------|-------------|---------|
| Accuracy  | 0.7551              | 0.7727      | +0.0176 |
| Precision | 0.8726              | 0.7901      | -0.0825 |
| Recall    | 0.6122              | 0.7586      | +0.1464 |
| F1-score  | 0.7196              | 0.7740      | +0.0544 |
| AUROC     | 0.9033              | 0.8582      | -0.0451 |
| MCC       | 0.5394              | 0.5460      | +0.0066 |

### Quality of the Predicted Probabilities

Figure 6 plots the observed disease rate against the predicted probability across 10 equal bins. The curve lies above the diagonal in every bin, indicating the model is consistently too cautious on this cohort: records it scores at 0.16 turn out to be positive 45 percent of the time, and records it scores at 0.34 turn out to be positive 66 percent of the time. The average predicted probability is 0.336, while the true rate in the external set is 0.513.

The cause is not a network fault but a difference between the two populations. The external cohort contains a higher share of sick patients than the corpus the model was trained on, so probabilities learned on one population are systematically too low on the other. This is also why the tuned threshold of 0.27 sits so far

below 0.5 and why the threshold step is not an optional extra in a cross-cohort study. A simple prior correction, in which the predicted odds are rescaled by the ratio of the two disease rates, or a light recalibration on a small, labeled sample from the new site, would remove most of this shift and improve the Brier score without retraining the network. Both are left for future work because neither can be applied without touching data from the target cohort, which this study deliberately avoided.



**Figure 6:** Reliability curve of ARO-FTAMNet on the external test set, showing observed disease rate against predicted probability.

**Figure Credit:** Rinaben Keshavlal Patel.

## DISCUSSION

Three points follow from these results. First, the numbers reported here are lower than those typically reported in the heart-disease literature, where accuracies above 95 percent are common. That difference is the point, not a weakness. Those figures come from testing on the same cohort used for training, often after oversampling. An accuracy of 0.7727 and an MCC of 0.5460 on records from a different source reflect what a user should expect, and the collapse of the three deep baselines in Table 6 shows, across two evaluation types, how large the difference between the groups exists.

Second, model size does not drive performance on tabular clinical data. FTAMNet is a small network, and the three baselines it beats have comparable or greater

capacity. What separates them is the input subset, the fitness score used during the search, and the tuned threshold. This aligns with broader evidence that for tabular problems with a few thousand rows, careful preparation and tuning matter more than depth<sup>14</sup>.

Third, the balance between the two error types should be set by clinical use, not by a default value in a library. The threshold analysis in Section 6.4 shows that the same trained weights can miss 127 or 265 patients, depending solely on the cutoff setting. Any deployment of a screening model should therefore report the threshold, explain how it was chosen, and present the confusion matrix, all of which are missing from much of the published literature.

### Limitations of this Study

The study has clear limitations. The external test uses a single cohort of 1,025 records, so the results describe transfer to that population, not to all populations; a broader study across several sites would be needed to support a general claim. The clinical fields available in the shared corpus are standard, so the model has no access to imaging, laboratory panels, or repeat measurements. The records are cross-sectional, so the model estimates current risk rather than future events. The probabilities are not calibrated for the target cohort, as Section 6.6 shows. Finally, the results come from a single training run; repeating the entire pipeline with several random seeds and reporting the spread would provide a firmer picture of stability.

### Future Research Recommendation

Future research will focus on validating ARO-FTAMNet using larger, multi-center, prospective cardiovascular datasets to assess its robustness and generalizability across diverse patient populations. The framework can be further extended to integrate multimodal clinical information, including ECG signals, medical imaging, laboratory measurements, and longitudinal electronic health records, to improve the comprehensiveness of cardiovascular risk assessment. Further investigation of multi-objective optimization can simultaneously address predictive performance, feature reduction, computational efficiency, and model complexity. At the same time, advanced explainability, calibration, and fairness analyses can enhance clinical reliability and interpretability. In addition, privacy-preserving federated learning and model compression techniques can be explored to support the secure and efficient

deployment of ARO-FTAMNet in real-world clinical decision support and resource-constrained healthcare environments.

## CONCLUSION

This paper presented ARO-FTAMNet, a heart-disease prediction framework in which Artificial Rabbits Optimization chooses the input attributes and the network settings in one search, a small gated multi-scale network makes the prediction, and the decision threshold is tuned on validation data and then frozen. The framework was trained on a hybrid corpus of 2,283 records and tested once on 1,025 records from a separate source.

On the unseen records, the model achieved an accuracy of 0.7727, a precision of 0.7901, a recall of 0.7586, an F1-score of 0.7740, an AUROC of 0.8582, an AUPRC of 0.8617, and an MCC of 0.5460. Three deep baselines trained on the same data failed to transfer, with MCC values ranging from -0.005 to 0.078. Removing the optimizer increased precision and AUROC but reduced recall by about 15 percentage points, which is the wrong direction for a screening tool. The predicted probabilities were systematically low on the external cohort because its disease rate is higher than that of the training corpus. A simple prior correction would address this shift.

The practical message is short. A model for clinical use should be evaluated on data it has never seen, scored with measures robust to class imbalance, and report the operating threshold. Future work will test the framework on several external cohorts simultaneously, add a prior-correction step to ensure the probabilities remain usable as the disease rate changes, repeat the pipeline across random seeds to assess stability, and add feature attribution so that each prediction can be explained to the clinician who must act on it.

### Consent for publication

The author has reviewed and approved the final version and agrees to be accountable for all aspects of the work, including any issues with accuracy or integrity.

### Disclosure

The authors declare that they have no financial involvement or affiliations with any organization, association, or entity, directly or indirectly related to the subject matter or materials presented in this review paper.

## Data availability statement

The data will be available only for research purposes.

## Authorship contribution

All authors contributed significantly to the work, whether in the conception, design, utilization, collection, analysis, or interpretation of data, or in all these areas. They also participated in drafting, revising, or critically reviewing the paper, gave their final approval of the version to be published, decided on the journal to which the article would be submitted, and made the responsible decision to be accountable for all aspects of the work.

## REFERENCES

- World Health Organization. Cardiovascular diseases (CVDs), WHO Fact Sheets, Geneva, Switzerland, 2021. Available from [https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)) [Accessed August 22, 2026]
- Roth GA, Mensah GA, Johnson CO, Addolorato G, Ammirati E, Baddour LM, Barengo NC, Beaton AZ, Benjamin EJ, Benziger CP, Bonny A, Brauer M, Brodmann M, Cahill TJ, Carapetis J, Catapano AL, Chugh SS, Cooper LT, Coresh J, Criqui M, DeCleene N, Eagle KA, Emmons-Bell S, Feigin VL, Fernández-Solà J, Fowkes G, Gakidou E, Grundy SM, He FJ, Howard G, Hu F, Inker L, Karthikeyan G, Kassebaum N, Koroshetz W, Lavie C, Lloyd-Jones D, Lu HS, Mirijello A, Temesgen AM, Mokdad A, Moran AE, Muntner P, Narula J, Neal B, Ntsekhe M, Moraes de Oliveira G, Otto C, Owolabi M, Pratt M, Rajagopalan S, Reitsma M, Ribeiro ALP, Rigotti N, Rodgers A, Sable C, Shakil S, Sliwa-Hahnle K, Stark B, Sundström J, Timpel P, Tleyjeh IM, Valgimigli M, Vos T, Whelton PK, Yacoub M, Zuhlke L, Murray C, Fuster V; GBD-NHLBI-JACC Global Burden of Cardiovascular Diseases Writing Group. Global Burden of Cardiovascular Diseases and Risk Factors, 1990-2019: Update From the GBD 2019 Study. *J Am Coll Cardiol*. 2020;**76**(25):2982-3021. doi: 10.1016/j.jacc.2020.11.010.
- Ali MM, Paul BK, Ahmed K, Bui FM, Quinn JMW, Moni MA. Heart disease prediction using supervised machine learning algorithms: Performance analysis and comparison. *Comput Biol Med*. 2021;**136**:104672. doi: 10.1016/j.combiomed.2021.104672.
- Mohan S, Thirumalai C, Srivastava G. Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques. *IEEE Access*. 2019; **7**: 81542-81554. doi: 10.1109/ACCESS.2019.2923707.
- Chen T, Guestrin C. XGBoost: A scalable tree boosting system. KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016. 785–794. Available from <https://dl.acm.org/doi/10.1145/2939672.2939785#:~:text=https%3A//doi.org/10.1145/2939672.2939785> [Accessed August 22, 2026]
- Prokhorenkova L, Gusev G, Vorobev A, Dorogush AV, Gulin A. CatBoost: Unbiased boosting with categorical features. *ArXiv*. 2019. Available from <https://arxiv.org/abs/1706.09516> [Accessed August 22, 2026]
- Dutta A, Batabyal T, Basu M, Acton ST. An efficient convolutional neural network for coronary heart disease prediction. *Expert Syst Appl*. 2020; **159**: 113408. Available from <https://doi.org/10.1016/j.eswa.2020.113408> [Accessed August 22, 2026]
- Shrivastava PK, Sharma M, Sharma P, Kumar A. HCBiLSTM: A hybrid model for predicting heart disease using CNN and BiLSTM algorithms. *Meas Sens*. 2023; **25**: 100657. Available from <https://doi.org/10.1016/j.measen.2022.100657> [Accessed August 22, 2026]
- Mirjalili S, Mirjalili SM, Lewis A. Grey Wolf Optimizer. *Adv Eng Softw*. 2014; **69**: 46-61. Available at <https://doi.org/10.1016/j.advengsoft.2013.12.007> [Accessed August 22, 2026]
- Rajammal RR, Mirjalili S, Ekambaram G, Palanisamy N. Binary Grey Wolf Optimizer with Mutation and Adaptive K-nearest Neighbor for Feature Selection in Parkinson's Disease Diagnosis. *Knowl-Based Syst*. 2022; **246**: 108701. Available from <https://doi.org/10.1016/j.knsys.2022.108701> [Accessed August 22, 2026]
- Wang L, Cao Q, Zhang Z, Mirjalili S, Zhao W. Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. *Eng Appl Artif Intell*. 2022; **114**: 105082. Available from <https://doi.org/10.1016/j.engappai.2022.105082> [Accessed August 22, 2026]
- Ali L, Rahman A, Khan A, Zhou M, Javeed A, Khan JA. An Automated Diagnostic System for Heart Disease Prediction Based on  $\chi^2$  Statistical Model and Optimally Configured Deep Neural Network. *IEEE Access*. 2019; **7**: 34938-34945. doi: 10.1109/ACCESS.2019.2904800.
- Latha CBC, Jeeva SC. Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques. *Inform Med Unlocked*. 2019;**16**: 100203. Available from <https://doi.org/10.1016/j.imu.2019.100203> [Accessed August 22, 2027]

14. Shwartz-Ziv R, Armon A. Tabular data: Deep learning is not all you need. *Inf Fusion*. 2022; **81**: 84-90. Available from <https://doi.org/10.1016/j.inffus.2021.11.011> [Accessed August 22, 2026]
15. Holland JH. Adaptation in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. 255 Main Street, 9<sup>th</sup> Floor, Cambridge, MA 02142, USA, The MIT Press, 1992. Available from <https://doi.org/10.7551/mitpress/1090.001.0001> [Accessed August 22, 2026]
16. Matthews BW. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim Biophys Acta*. 1975; **405**(2):442-51. doi: 10.1016/0005-2795(75)90109-9.
17. 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**(1):6. doi: 10.1186/s12864-019-6413-7.
18. Saito T, Rehmsmeier M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS One*. 2015; **10** (3):e0118432. doi: 10.1371/journal.pone.0118432.
19. Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. ICML'15: Proceedings of the 32nd International Conference on Machine Learning. 2015; **37**: 448-456. Available from <https://dl.acm.org/doi/10.5555/3045118.3045167> [Accessed August 22, 2026]
20. Ba JL, Kiros JR, Hinton GE. Layer Normalization. *ArXiv*. 2016. Available from <https://arxiv.org/abs/1607.06450> [Accessed August 22, 2026]
21. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). *ArXiv*. 2016; 770-778. Available from <https://arxiv.org/abs/1512.03385> [Accessed August 22, 2026]
22. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting. *J Mach Learn Res*. 2014; **15**: 1929-1958. Available from <https://dl.acm.org/doi/epdf/10.5555/2627435.2670313> [Accessed August 22, 2026]
23. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems. 6000-6010. Long Beach, California, USA, 2017. Available from <https://dl.acm.org/doi/proceedings/10.5555/3295222> [Accessed August 22, 2026]
24. Kingma DP, Ba J. Adam: A method for stochastic optimization. The 3rd International Conference for Learning Representations, San Diego, California, USA. *ArXiv*. 2015. Available from <https://doi.org/10.48550/arXiv.1412.6980> [Accessed August 23, 2026]
25. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: Synthetic minority over-sampling technique. *J Artif Intell Res*. 2002; **16**: 321-357. Available from <https://doi.org/10.48550/arXiv.1106.1813> [Accessed August 22, 2026]
26. Krittanawong C, Virk HUH, Bangalore S, Wang Z, Johnson KW, Pinotti R, Zhang H, Kaplin S, Narasimhan B, Kitai T, Baber U, Halperin JL, Tang WHW. Machine learning prediction in cardiovascular diseases: a meta-analysis. *Sci Rep*. 2020; **10**(1):16057. doi: 10.1038/s41598-020-72685-1.
27. 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. *Am J Cardiol*. 1989; **64**(5):304-10. doi: 10.1016/0002-9149(89)90524-9.
28. Brier GW. Verification of forecasts expressed in terms of probability. *Mon Weather Rev*. 1950; **78** (1): 1-3. Available from [https://doi.org/10.1175/1520-0493\(1950\)078<0001:VOF EIT>2.0.CO;2](https://doi.org/10.1175/1520-0493(1950)078<0001:VOF EIT>2.0.CO;2) [Accessed August 22, 2026]