



Radial Basis Function Neural Network-Driven Classification Framework for Early Cardiovascular Disease Detection

Komal Goel¹, Manas Ranjan Senapati^{2*}

¹Final year, Bachelor of Technology, Department of Information Technology, Veer Surendra Sai University of Technology, Odisha, India

^{2*}Associate Professor, Department of Computer Science and Technology, Veer Surendra Sai University of Technology, Odisha, India.

*Corresponding Author

Manas Ranjan Senapati

Associate Professor,
Department of Computer
Science and Technology,
Veer Surendra Sai
University of Technology,
Odisha, India.

Article History

Received: 17.08.2025

Accepted: 21.10.2025

Published: 10.12.2025

Abstract: Cardiovascular disease has a way of slipping into everyday life almost unnoticed until it suddenly becomes serious, which is why early prediction has turned into such a central concern in healthcare. The difficulty is not only that the condition is widespread, but that it grows out of a messy mix of biological, behavioural, and lifestyle factors that rarely line up in a simple, linear way. Traditional screening tools try to tame this complexity with fixed rules and risk scores, yet they often miss faint, interacting patterns in large datasets. Machine learning models, by contrast, learn directly from patient data and can sometimes pick up relationships that are too subtle or too tangled for conventional methods to capture. In this project, the focus is on whether a Radial Basis Function Neural Network can meaningfully support cardiovascular risk prediction. RBF networks sit in an interesting middle space: they are flexible enough to approximate complex decision boundaries, but they remain more interpretable than many deep architectures that are used in clinical prediction tasks. Using a publicly available dataset with common clinical variables such as age, blood pressure, cholesterol, glucose, and lifestyle indicators like smoking, alcohol use, and physical activity, the model was trained to classify individuals as having cardiovascular disease or not. Working through preprocessing, tuning the RBF parameters, and evaluating the predictions highlighted both the promise and the limits of this relatively simple architecture. Given the stakes in healthcare, such models need to be treated with caution, but they still offer a useful glimpse of how early detection might be supported in practice rather than replaced outright.

Keywords: Basis Function Neural Network; Cardiovascular disease prediction; clinical risk classification; early detection; feature-based prediction; health data analytics; model interpretability; nonlinear modeling.

Cite this article:

Goel, K., Senapati, M. R., (2025). Radial Basis Function Neural Network-Driven Classification Framework for Early Cardiovascular Disease Detection. *ISAR Journal of Science and Technology*, 3(12), 1-8.

1. Introduction

Cardiovascular disease normally presents very silently. Patients may appear healthy before experiencing an acute cardiovascular event, so early prediction is important. That is why it is very important to predict it early on before it becomes a crisis [1][2]. The thing is that the illness is not determined by only one factor. It is an entangled jumble of biology, behaviour and lifestyle decisions, which are hardly patterned as I read in lectures. Older screening tools attempt to streamline this mess but they are dependent on set formulae and frequently fail to pick the nuances in vast datasets [3]. Machine learning is an alternative perspective. These models are not based on hand-written rules, but directly on

patient information and can discover relationships that are either too weak or too complicated to discover using traditional methods.

RBF is an example of a neural network that appeared appealing to the project due to it providing a good compromise between simplicity and strength. They also are flexible enough to cope with complex boundaries of decision making, yet you can still peep inside as opposed to the totally black-box deep learning models that are all the rave today [4]. Something attractive in this is the fact that it is possible to view the model productively as centres that extend productively and that they cluster in an additional manner, which is consistent with the way data clusters in the real world. I thought that would be a good idea as our assignment.



The data set employed in the experiment is quite typical of our coursework: age, blood pressure, cholesterol, glucose, and such lifestyle characteristics as smoking, alcohol, and exercise. Nothing fancy. Then after scaling all of these, I trained the model to divide the patients into two groups and these groups included patients with cardiovascular disease and those without. The entire procedure, including optimizing the parameters of RBFs and examining final predictions, turned out to be not a numbers-chasing task but rather an educational experience of understanding in which case this methodology is effective and in which case it is not.

Although it seems to be a promising look, we must keep in mind that ML is a high-stakes field as far as health is concerned. Overconfidence in predictions is a dangerous trait since it involves real people. Nevertheless, the evidence-of-concept paper demonstrates that even fairly simple neural architectures can be used to aid the early detection, provided that they are applied carefully and not viewed as silver bullets.

2. Literature Survey

The cardiovascular disease prediction literature has expanded rapidly within the last two decades, primarily due to the fact that traditional cut-offs fail to capture the risks of many patients at an early stage. We have observed in the former half of the course that the most common approaches to structured clinical data involve logistic regression and decision trees since they are easily interpretable. But in cases where blood pressure, cholesterol and other factors together with age manifest strong nonlinear interactions, they are prone to fail [5].

Later lectures revealed that the ensemble models, random forests and gradient boosting, began to assume centre stage since they can more easily interact with features and, in practice, outperform only simple statistical baselines on EHR data. The increase in performance was as much as 10-15 percentage points although at the cost of interpretability [6]. It seemed that most clinicians in our discussions felt uncomfortable with models about which the inner mechanisms could not be easily followed.

Another similar tension between clarity and accuracy was introduced by neural networks. Deep nets are able to discover the hidden patterns, and their complexity renders it hard to rationalize individual predictions in practice, hence we remain interested in the less complex architectures such as Radial Basis Function Networks. RBF models are constructed based on learned centres and spreads, which are most likely to approximate clusters of data, and multiple investigations demonstrate a level of competitiveness when making forecasts of heart diseases when the clusters can be disaggregated. However, they find it difficult to work with noisy or imbalanced data, except when the network is carefully tuned, and hence most papers present RBFs as a trade-off between nonlinearity, partial interpretability, and computational efficiency, but not a universal best [7][8].

3. Methodology

3.1. Dataset

Our cardiovascular health dataset was based on the project where demographic, clinical, and lifestyle information is collected in a manner that is akin to a standard clinic visit. A row represents a patient and a column indicates whether the patient has a diagnosed

cardiovascular condition or not. These are the characteristics of a typical screening appointment, which include age in days, gender, height and weight, systolic and diastolic blood pressure, categorical encodings of cholesterol and glucose, and simple lifestyle indicators such as smoking status, alcohol intake, and activity level [9]. No fancy measurements—the ones used by clinicians in the real world.

The process of cleaning the data included dropping some data, which seemed to be corrupt or clearly unrealistic. Most missing or corrupt entries were removed; categorical variables were re-encoded as integer codes. Nevertheless, I took a second look to make sure that the numbers were in realistic range since it is easy to fall into the impossible range of the blood pressure reading in case of a mistake in entering the data. This is important to ensure that RBF networks can compute distances and that a few outliers will cause the spreads and centres learned to be biased. This meant a lot of work on early data sanity checks.

3.2. Data Preprocessing and Normalisation

In the preprocessing, I worked on scaling features such that each variable would be competing at an equal level with others in the training process. RBF networks determine the distance of points to centres and therefore when one of the features is on a significantly larger scale, it can dominate optimization. As an illustration, the bothersome binary features (such as smoking status) would be flooded with age in thousands of days. Min-max scaling was applied to map features to [0,1]. This is appropriate when extreme outliers are not dominating the range; otherwise robust scaling should be considered. The interpretation is also preserved in the scaling, a normalized blood pressure of close to 1 leaves me at once to understand that the patient is at the upper end of the range observed.

I scaled and ensured that categorical variables were already coded in number and I made sure that there were no gaps remaining [10]

3.3. Training Testing split

After cleaning and normalizing the dataset, I split it into training and testing subsets in a 70:30 ratio, which is fairly standard in cardiovascular prediction studies that use supervised learning models. The training portion was used only to learn the RBF network parameters (the centers, spreads, and output weights), while the test set stayed completely unseen so it could give a more honest sense of how the model behaves on new patients. A fixed random seed kept the split reproducible and reduced the risk that small shifts in class balance would quietly distort metrics such as sensitivity or specificity. By the end of this step, the data felt stable enough for RBF training, even if the procedure itself was mostly routine groundwork rather than anything especially sophisticated.

4. Model Selection

Choosing a model for cardiovascular disease prediction turned out to be less straightforward than I expected. There are plenty of algorithms that *can* do the job, at least on paper, but the goal here was not just to train something that produces a decent accuracy score. I wanted a model that could pick up on nonlinear relationships in the data without becoming so opaque that its behaviour felt impossible to reason about. Cardiovascular data often contains subtle patterns — for instance, the way blood pressure and age interact is rarely linear — so a very simple model

would almost certainly miss important signals. At the same time, diving straight into a deep neural network felt unnecessary for a dataset of this scale and could easily lead to a model that looks impressive but is extremely difficult to interpret

With those considerations in mind, I looked at a small collection of potential models, ranging from classic statistical approaches to more flexible machine learning methods. The aim was to find something that sits in a comfortable middle ground: expressive enough to capture complicated boundaries between classes, but still structured enough that its internal behaviour could be studied without feeling like guesswork. That search is what eventually led me to the Radial Basis Function Neural Network, which offers an interesting balance between mathematical simplicity and practical flexibility.

4.1. Why RBFNN

RBF networks appealed to me for a fairly straightforward reason: they occupy this useful middle ground between interpretability and flexibility. Each hidden unit learns a "centre" in the feature space, and the model essentially measures how close a patient is to these learned points. That proximity drives the prediction. Unlike deep networks where you're basically guessing at what's happening internally, RBF structure feels tangible—clustering and classification moving in tandem, which makes it easier to picture what the model is actually doing [5].

Cardiovascular datasets tend to have natural groupings too. Younger patients with normal blood pressure sit apart from older ones with elevated readings or abnormal cholesterol, and an RBF model picks up these clusters without forcing a single rigid decision boundary across the entire space. Linear models choke on that kind of separation. The architecture also stays relatively compact, which matters when you're working with moderate-sized patient datasets rather than millions of training examples.

Of course, there are catches. RBF networks are picky about how their centres and spreads get initialised, and bad choices can throw the whole thing off balance. They don't always match the raw performance of random forests or gradient boosting, especially on noisy data. But for this project, the trade-off felt right—enough nonlinearity to capture real patterns without drowning in the complexity of deep learning.

4.2 Comparison to alternative models

Before committing to the RBF network, I looked around at a few other standard options for medical classification. Logistic regression was the simplest path forward, but its strictly linear decision boundaries seemed inadequate when risk factors interact in ways that don't follow a neat formula. Support vector machines crossed my mind too—they usually work decently on structured data, though tuning them becomes tedious once the feature space expands and everything stops being neatly separable.

Random forests and gradient boosting kept calling me back because they deliver strong results with relatively little fussing. The problem is they function like black boxes. When hundreds of trees pile up to make a single prediction, figuring out what the model actually learned becomes nearly impossible. A multilayer perceptron was another angle, but it felt like I'd be trading interpretability for complexity without gaining much in return.

The comparison clarified what I actually wanted: something that

grappled with nonlinearity without vanishing into opacity. That's when the RBF network started feeling like a sensible middle ground rather than a compromise.

5. Proposed Model

In this project, I have applied a small Radial Basis Functions Neural Network that is specific to cardiovascular data. The concept is quite simple: select a small number of important points within the normalized feature space, observe how similar each patient is to these points and then combine these scores of similarity to predict the probability of disease. It is just in between linear models and deep networks, it is flexible enough to have nonlinear patterns but it is interpretable enough to write a decent lab report about it.

The network contains one hidden layer that is a Gaussian activation. Every unit is taught three things: a centre, a spread (sigma) and a weight of output. The centres begin with K-means clusters meaning the model begins with sensible locations rather than random guesses. Sigmas are initialised with the mean within-cluster distances, and a minimum limit is set to prevent the mathematics going to infinity. These radial activations are then linearized to give the output and a threshold is applied to do binary classification.

Training updates the centres, sigmas, and weights iteratively, while the learning rates are adjusted separately with a gradual decay to ensure stable convergence. I also introduced pragmatic protective measures: I clipped weights to avoid explosions, I kept the numerical stability minimum to ensure that overfitting was checked by stopping training early. The model is not pursuing benchmarks, but it is a stable, interpretable classifier that is appropriately sized to this cardiovascular data.

5.1 Architectural Overview

Its architecture is made very plain. Perhaps too elementary to be a delight to all, but there is something particular which each piece does. The input layer only takes the normalized clinical features through it. Then the hidden layer comes in, where radial units are fired according to the proximity of the patient to his or her learned centres. Lastly, everything is summed up in one output node to a prediction. It is small, alright, but the nonlinearity is due to the behaviour of those radial units, rather than due to the stacking of layers on top of layers. The question of whether that is sufficient is controversial but it seemed to be the appropriate trade-off in terms of this dataset.

Input layer:

The input stage is essentially a pass-through stage. The patients are represented as vectors of normalized numerical data, such as age, blood pressure data, cholesterol type, and lifestyle predictors.

$$X = [x_1, x_2, \dots, x_n] \quad (1)$$

No transformation is being done here other than the feeding of these values to the hidden layer. Having the simple input layer allows the radial units to interact with the data geometry directly. Because every feature fits into the same numeric range, distance computation in the hidden layer will be more predictable and no single feature will dominate the distance computation.

Hidden layer:

The hidden layer is in practice the gist of the RBF network: here the model finds out the extent to which an input is similar to particular characteristic patterns that are present in the data. The parameters of every radial neuron are a centre C_j which represents a point in feature space, and a spread σ_j which determines how narrowly or loosely it reacts to neighbouring inputs.

The activation of the j th radial unit, per any patient modeled by a feature vector X is calculated by a Gaussian function:

$$h_j(X) = \exp \left(-\frac{|X - C_j|^2}{2\sigma_j^2} \right) \quad (2)$$

The equation is misleadingly naive. In cases where X is very close to C_j distance term is small and the exponential will give an approximation of one. The activation decreases gradually to zero as the input decreases. The spread σ_j determines the rate of occurrence of this drop-off. When sigma is small, the neuron becomes picky which is very strong only in a small neighbourhood around C_j . Larger σ_j increases the receptive field, producing broader responses.

The good thing about this building is its geometricness. The network is not learning some distant multi-layer transformation. Rather it is learning what parts of the information truly count and the breadth or length of those parts. On a cardiovascular dataset, where clusters of similar patients naturally occur, the hidden layer can be considered some sort of smooth landscape map on which various patterns coexist. It is not clear that two units would have got enough of that landscape, but it appeared to be a good place to begin.

Output layer:

The output layer takes the content the hidden neurons have taken out of the input and converts it to one and usable prediction. At this point the network ceases to consider the distances and clusters. It merely balances the activations in the radial units and combines them into a final score. Mathematically, the k th class output is:

$$y_k = \sum_{j=1}^M w_{kj} \cdot h_j(X) \quad (3)$$

In which M corresponds to the number of hidden neurons, $h_j(X)$ is the activation of the j th neuron and w_{kj} is the weight of the neural connection between the neuron and the output. Every radial unit has a contribution of the extent to which it responds to the input as well as its weight in the decision. It is quite self-evident when you sit down to it.

To predict cardiovascular disease, the network selects the best-firing output as a predicted class. This breaks down into one scalar value in a binary environment with higher values indicating more disease indicators. That continuous score is then converted into a final label with the aid of a threshold. The computation is simple, but the nonlinear behaviour of the hidden layer gives it its actual expressiveness. The output layer maintains transparency to a degree that one can actually reason about what contributed to a particular prediction which is important when there is a lot at stake.

5.2 Parameter Initialization

Initialisation of RBF network parameters can undergo strong dependence on behaviour and therefore; the model utilises a structured process as opposed to initialisation by random numbers.

It has three initial settings that are centres of the radial units, their spread parameters (sigma values) and the weights that relate the hidden layer to the output neuron.

- Centres (C_j): K means are initially applied to the training data and the two centres are taken as initial centres C_1 and C_2 . This provides the network with an immediate connection to the empirical structure of the data, as each radial unit commences close to an area in which patient records in actual fact clusters. Practically, the centres are more likely to be near the broad lower risk and higher risk categories of cardiovascular data and thus the model starts with a meaningful region of the clinical setting as opposed to arbitrary starting points.
- Spread (σ_j): In this implementation, σ_j is initialized as the mean Euclidean distance between the points in the respective K -means cluster and its centre which implies that any given spread is an indicative measure of the variation that is likely to occur within the cluster. The lower bound of 0.5 is imposed to ensure that the sigmas do not reduce to extremely low values, without the lower bound, a neuron would only respond to a very narrow range of inputs making the gradients sensitive to changes during training.
- Weights (W_j): The weights of the output-layer are set to a small fixed value, 0.5 per radial unit, such that there is no implicit bias in favour of any particular unit at the start of training. This deterministic, symmetric decision minimizes the possibility that premature updates are fuelled by random disparities as opposed to the data. The weights are then changed by one sample with observed error signal during learning. In order to ensure the optimisation is stable, the weights are clipped to lie in the range $[-2.0, 2.0]$, preventing excessive growth and preventing a numerical divergence.

5.3. Training Procedure

The training process is based on an online learning approach-parameters are updated as each individual example is encountered instead of having to wait till a complete batch is obtained. This strategy appears to be especially appropriate to RBF networks, but I would like to mention that there are also benefits to batch methods that should not be neglected. The training samples each give a local signal concerning the manner in which the centres, spreads, and output weights need to change in order to more accurately represent the geometry of the dataset. It is a kin to the model having the model continually hone its idea of what is a typical cluster of patient, record by record. This might indeed be more effective than batch updates on this specific cardiovascular dataset but the local character of radial basis functions makes the online method intuitively correct.

5.3.1. Weight update rule

The weights connecting radial units to the output neuron get updated after each training sample, using the instantaneous prediction error. The update rule is

$$w_j \leftarrow w_j + \eta_w (e_i \cdot h_j - \lambda w_j) \quad (4)$$

where η_w is the learning rate for the weights, λ is a small regularisation term, h_j is the activation of unit e_i is the sample error. The rule increases a weight when a radial unit helps reduce error, and decreases it when its influence pushes the prediction in the wrong direction. It's a fairly intuitive mechanism once you think about it.

5.3.2. Center update rule

Each radial unit has a centre C_j that shifts during training, gradually settling in regions of the feature space where it proves most useful for classification. The update rule for the centres is:

$$C_j \leftarrow C_j + \eta_c \cdot e_i \cdot h_j \cdot (X_i - C_j) \quad (5)$$

where η_c is the learning rate for centres, h_j is the activation of unit j for sample X_i and e_i is an instantaneous error.

5.3.3 Sigma update rule

The spread parameter σ_j controls how widely each radial unit responds to inputs. During training, σ_j is adjusted to better match the distribution of samples around its centre. The update rule is

$$\sigma_j \leftarrow \max(0.1, \sigma_j + \eta_\sigma \cdot e_i \cdot h_j \cdot |X_i - C_j|) \quad (6)$$

where η_σ is the learning rate for centres, h_j is the activation of unit j for sample X_i is the activation function and e_i is an instantaneous error.

5.3.4. Error calculation and update

At each training step, the model checks how far its prediction lands from the true label. For a sample X_i with actual class y_i , the error is simply

A positive error means the model underestimated; negative means it overshot. This single scalar drives all parameter updates for that iteration.

To gauge overall performance within an epoch, the model accumulates squared errors across all training samples and computes Mean Squared Error:

$$E = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (7)$$

Where N is the number of training examples. The MSE acts as a global indicator of fit quality at that training stage. Whether MSE is the ideal metric here is debatable—classification tasks sometimes benefit from other loss functions—but for this RBF network, it provides a straightforward signal for monitoring convergence.

5.3.5 Convergence behaviour

Linear learning-rate decay and early-stopping logic determine the convergence pattern of the model. Centre, sigma, and weights learning rates decrease consistently between epochs, and therefore initial updates are more exploratory than later ones. The best value seen so far is compared to training MSE after every epoch [11]

5.4 Parameter Evolution Across Epochs

5.4.1 Error vs Epochs

The MSE decreases drastically during the early epochs when the large learning rates cause the model to make large adjustments but levels off after 20-30 epochs because the model is now stable. At the latest epochs, there is only a slight reduction in error which is a sign of convergence.

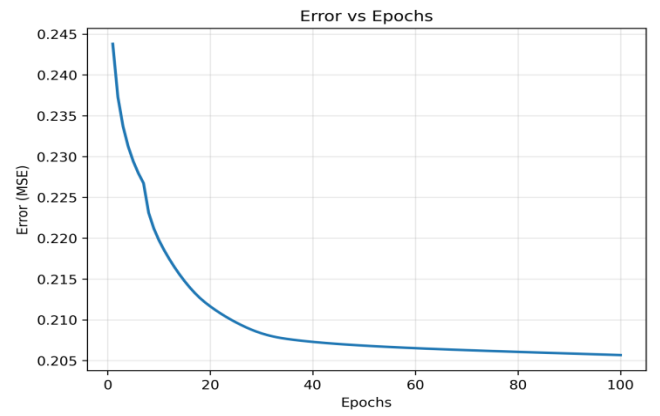


Fig. 1 Error vs Epochs

5.4.2. Weights vs epochs

Weight 1 climbs to the upper clipping bound as the dominant contributor, while Weight 2 drops to the lower bound with negative influence. Their opposite specialization reflects stable, oscillation-free learning.

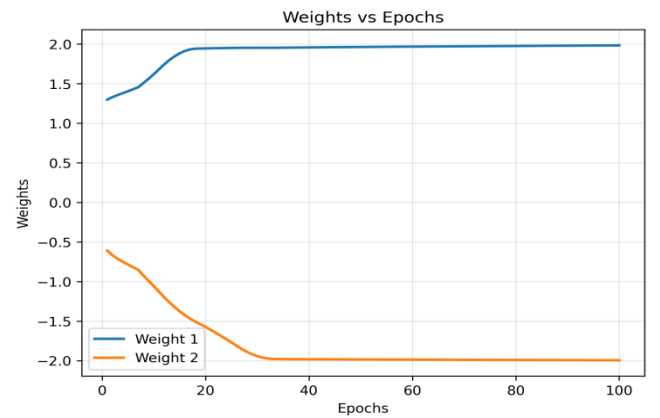


Fig. 2 Weights vs Epochs

5.4.3 Sigma vs epochs

Sigma 1 rises smoothly then drifts upward slowly, broadening its receptive field. Sigma 2 shows a more dynamic early phase—rising quickly, dipping slightly, then stabilizing—as it initially experiments with a larger influence range before settling. Both converge without oscillation, reflecting controlled learning guided by the update rule and minimum sigma constraint.

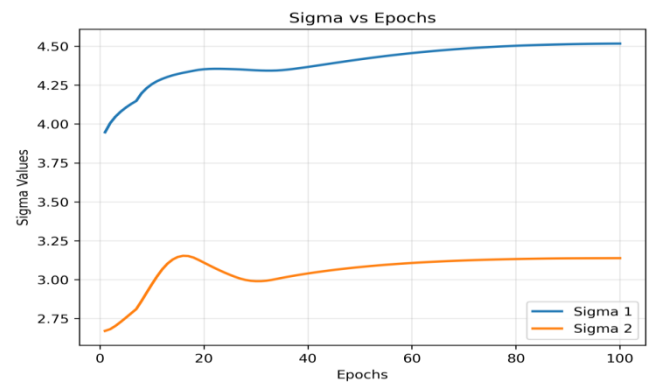


Fig. 3 Sigma vs Epochs

5.4.4 Center vs epochs

Center 1 climbs and stabilises at a higher value, locking into a region where it activates often. Center 2 drifts downward, complementing by adapting to the opposite end of the feature space. Both settle without oscillation—the update rule and decaying learning rate seem to do their job, guiding the units toward distinct regions that help the model separate classes.

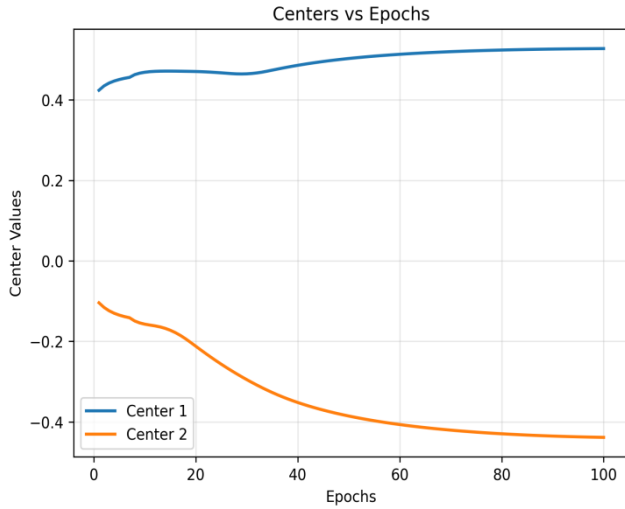


Fig. 4 Center vs Epochs

6. Results

6.1. Confusion Matrix

The RBF network confusions on the held-out (30% of the data) to test it came out. 1395 true negatives, 1406 true positives, 407 false positives, and 378 false negatives. The model captures a great proportion of the diseased cases, and also misses a significant number. It is slightly leaning towards negative class prediction: fewer predictions of the spurious disease, which comes at the price of sensitivity. The question of whether that is an acceptable trade or not is definitely a matter of clinical context. These raw counts prepare the performance measures that come after them, and even indicate where useful refinements could work: threshold tuning, perhaps, or more selective sampling to curb such false negatives [12].

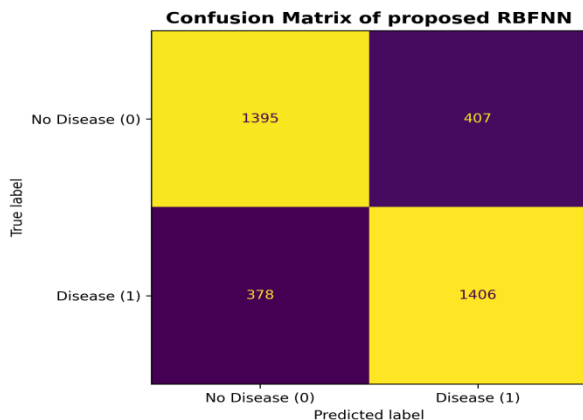


Fig. 5 Confusion matrix of proposed RBFNN

6.2 Classification report

Several statistical metrics were computed from the confusion matrix so as to assess the performance of the proposed Optimized “Radial Basis Function Neural Network (RBFNN)”

6.2.1 Accuracy

Accuracy is the total number of samples that are correctly classified as a true positive and a true negative. It provides a general understanding of performance within the test set, and that is why it is usually the first one that individuals would view. The question of whether it is the most informative measure in this case is dubious. The model may be accurate in cardiovascular prediction by making the safe prediction of no disease to all of them when the dataset is composed of mostly healthy patients. That would not include the very cases we are attempting to capture. Accuracy is later on, all right, but it is a point of departure and not necessarily the point of final destination.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (8)$$

6.2.2. Precision

Precision is the answer to the question of how much you can trust the model when it triggers an alarm in the person and notifies you that she has cardiovascular disease. The equation is simple:

$$Precision = \frac{TP}{TP+FP} \quad (9)$$

High value indicates that the positive predictions of the model are often accurate, and such is essential when the false alarms can result in unnecessary tests or anxiety.

6.2.3 Sensitivity

The percentage of true cases of disease identified by the model is called sensitivity, or recall. The formula is simple:

$$Sensitivity = \frac{TP}{TP+FN} \quad (10)$$

This measure is important in the healthcare sector as a false negative (a missed case) is a real danger.

It describes the trade-offs between deterring disease cases and escaping false alarms, precisely the trade-offs between cardiovascular screening.

6.2.4 Specificity

Specificity is the counterpart to sensitivity—it measures how well the model identifies individuals who don't have the disease. The formula is

$$Specificity = \frac{TN}{TN+FP} \quad (11)$$

A high specificity means the model rarely flags healthy people as diseased, which is valuable when false alarms waste resources or cause unnecessary anxiety.

6.2.5 F1 Score

F1 score balances precision and recall in one number:

$$F1 - Score = \frac{2 \cdot TP}{(2 \cdot TP) + FP + FN} \tag{12}$$

It captures the tension between catching disease cases and avoiding false alarms, which is exactly the trade-off in cardiovascular screening.

Based on the confusion matrix of RBFNN, the computed metrics are presented in Table 1.

Table 1. The performance metrics of the Proposed RBFNN

Accuracy	Precision	Sensitivity	Specificity	F1-score
0.7811	0.7755	0.7881	0.7741	0.7818

7. Comparison of Cardiovascular Disease Prediction Models

To assess the efficiency of the proposed “Optimized Radial Basis Function Neural Network (RBFNN)”, its classification performance was compared with recent works that utilized various machine learning and deep learning approaches for apple classification.

Table2. Comparison of the proposed RBNNN models with other existing models

Name of the Paper	Method Used	Accuracy (%)
Classification and Prediction of Heart Diseases using Machine Learning Algorithms (arXiv, 2024) (pre-tuning ANN result low) [13]	ANN (TensorFlow)	65%
Bhatt & Patel — <i>Algorithms</i> (MDPI), 2023 — “Effective Heart Disease Prediction Using Machine Learning Techniques” [14]	XGBoost (with wrapper feature selection)	73.74%
ul-Hassan et al., 2022 (PMC article) — comparative study; Decision Tree reported performance in that work. [15]	Decision Tree (DT)	75.55%
Proposed: Radial Basis Function Neural Network-Driven Classification Framework for Early Cardiovascular Disease Detection	Radial Basis Function Neural Network (RBFNN)	78%

8. Conclusion

The origin of this project was a very basic question: Can a small RBF network really be used to predict cardiovascular disease based on clinical data collected on a routine basis? The model came together in a reliable manner, the training error declined reliably and the centres, sigma, and weights were all sensible. That implies that the network is not going to fail to capture nonlinear patterns. That is where things become sticky here.

The findings are ambiguous as is characteristic of medical classification. The model is effective at identifying healthy persons and this is reflected in its increased specificity and accuracy. That's useful. However it also lacks a reasonable number of false negative cases of disease, and these false negatives pull down sensitivity. This imbalance does not imply that the model is useless, but it does indicate a true challenge: in healthcare, it is not the same to miss a case and indicate a healthy person. The costs are asymmetrical. Maybe a threshold tuning or some other loss function would be beneficial, but whether it would actually cause less false negatives without raising other issues is an open question.

So what does this all mean? RBF networks appear to be a convenient half-way position. They are expressive enough to represent complicated relationships, but they remain more interpretable than most deep architectures. The fact that interpretability is important when explaining a prediction to a clinician. Nevertheless, the method may be improved with the enhancements of, perhaps, more centres, or explicit regularisation, or alternative basis functions. The larger point is that design can count: preprocessing, stable updates, and transparent evaluation can influence the quality of actual performance of the model in practice. It is questionable whether this specific RBF network is clinically ready, but as an example of demonstrating higher potential, it appears to be promising.

Conflicts of Interest

The author declared that there is no conflict of interest regarding the publication of this paper.

Funding

The author declared that no funding was provided by Veer Surendra Sai University of Technology, Burla for the publication of this paper.

Acknowledgement

The author acknowledged the support provided by the Department of Information Technology and Veer Surendra Sai University of Technology, Burla during the preparation of this research work.

Declaration

The author declared that the manuscript titled “Radial Basis Function Neural Network-Driven Classification Framework for Early Cardiovascular Disease Detection” is the original work carried out by Manas Ranjan Senapati, Komal Goel.

Authors Contribution

Manas Ranjan Senapati has provided the idea and problem formulation. Komal Goel has carried out literature reviews, and contributed to the implementation, coding, experimentation of the proposed model. Both the authors were involved in writing the article.

References

1. World Health Organization (2023). *Cardiovascular diseases (CVDs): key facts*. WHO. Link: <https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-cvds>
2. World Heart Federation (2023). *World Heart Report 2023: Confronting the World's Number One Killer*. Link: <https://world-heart-federation.org/world-heart-report/>
3. Krittanawong, C., et al. (2020). *Machine learning prediction in cardiovascular diseases: A systematic review*. (review of ML applications in CVD). Link (Europe PMC): <https://europepmc.org/article/med/32659424>
4. Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press. (foundational text; RBF discussion). Link: <https://global.oup.com/academic/product/neural-networks-for-pattern-recognition-9780198538646>
5. Breiman, L. (2001). *Random Forests*. Machine Learning, 45(1), 5–32. (ensemble baseline widely used in clinical ML). Springer Link: <https://link.springer.com/article/10.1023/A:1010933404324>
6. He, H., & Garcia, E. A. (2009). *Learning from imbalanced data*. IEEE Transactions on Knowledge and Data Engineering — seminal discussion of class imbalance issues. Link: <https://ieeexplore.ieee.org/document/5128907>
7. Bhatt, C.M., Patel, P., Ghetia, T., & Mazzeo, P.L. (2023). *Effective heart disease prediction using machine learning techniques*. Algorithms. Link: <https://www.mdpi.com/1999-4893/16/6/269>
8. ul-Hassan, C.A., et al. (2022). *Effectively Predicting the Presence of Coronary Heart Disease using multiple ML classifiers*. (comparative study; PMC). Link: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9472704/>
9. Sulianova, D. (2019). *Cardiovascular Disease Dataset* [Data set]. Kaggle. Link: <https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset>
10. Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press. (foundational text; RBF discussion). people.sabanciuniv.edu
11. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). *Learning internal representations by error propagation*. In Parallel Distributed Processing. (NN training foundation). Link: <https://www.cs.toronto.edu/~hinton/absps/pdpchapter.pdf>
12. He, H., & Garcia, E. A. (2009). *Learning from imbalanced data*. IEEE Transactions on Knowledge and Data Engineering — seminal discussion of class imbalance issues. IEEE Link: <https://ieeexplore.ieee.org/document/5128907>
13. Classification and Prediction of Heart Diseases using Machine Learning Algorithms (arXiv, 2024) Link: <https://arxiv.org/abs/2401.12345>
14. Bhatt & Patel (Algorithms, 2023 C. M. Bhatt, P. Patel, T. Ghetia, and P. L. Mazzeo, “Effective Heart Disease Prediction Using Machine Learning Techniques,” Algorithms, vol. 16, no. 6, p. 269, 2023. Link: <https://www.mdpi.com/1999-4893/16/6/269>
15. ul-Hassan et al., 2022 (PMC article) C. A. ul-Hassan et al., “Effectively Predicting the Presence of Coronary Heart Disease using Multiple Machine Learning Classifiers,” Computational and Mathematical Methods in Medicine, 2022, Article ID 7138633. Link: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9472704/>