International Journal of Multidisciplinary Research and Development
www.allsubjectjournal.com
ISSN Online: 2349-4182
Received: 03-08-2026, Accepted: 02-09-2026, Published: 17-09-2026
Volume 13, Issue 3, 2026, Page No. 250-257

TPOT-RBFN: A hybrid genetic programming and radial basis function network approach for enhanced predictive analytics

Lim Eng Aik*
Faculty of Intelligent Computing, Department of Mathematical Sciences, Universiti Malaysia Perlis, Arau, Perlis, Malaysia
Corresponding Author: Lim Eng Aik
DOI: https://doi.org/10.66856/ijmrd.2026.13.3.13380

Abstract
This paper proposed TPOT-RBFN, a novel hybrid approach that integrates Tree-based Pipeline Optimization Tool (TPOT) with Radial Basis Function Networks (RBFN) to address the challenges of automated predictive analytics. The methodology combines the automated pipeline optimization capabilities of TPOT, which employs genetic programming to select optimal preprocessing steps and model configurations, with the robust function approximation properties of RBFN, a neural network architecture known for its ability to model complex data patterns. The RBFN component is characterized by its hierarchical structure, where radial basis functions in the hidden layer enable non-linear transformations, while the output layer linearly combines these transformations to produce predictions. The TPOT framework optimizes critical RBFN hyperparameters, including the number of basis functions, their centers, and widths, as well as the weights connecting the hidden and output layers, thereby enhancing the model's predictive accuracy. This integration not only automates the traditionally manual and error-prone process of model selection and hyperparameter tuning but also leverages the complementary strengths of genetic programming and neural networks to handle high-dimensional and non-linear datasets effectively. Experimental results demonstrate that TPOT-RBFN outperforms conventional methods in terms of both accuracy and computational efficiency, making it a scalable solution for real-world predictive analytics tasks. The proposed method is particularly significant for domains requiring interpretable yet powerful models, as it balances automation with performance while reducing the reliance on domain expertise. Furthermore, the flexibility of TPOT-RBFN allows it to adapt to diverse datasets, offering a generalizable framework for advancing predictive modeling in both academic and industrial applications.

Keywords: TPOT-RBFN, automated pipeline, genetic programming, hyperparameter tuning, predictive analytics

Introduction
Predictive analytics has become a cornerstone of modern data-driven decision-making across diverse domains, from healthcare diagnostics to financial forecasting. The field has witnessed significant advancements through the integration of machine learning techniques, yet challenges persist in achieving both high accuracy and computational efficiency, particularly when dealing with complex, high-dimensional datasets. Traditional approaches often require extensive manual intervention for model selection and hyperparameter tuning, which can be both time-consuming and suboptimal [1, 2].

Genetic Programming (GP) has emerged as a powerful evolutionary computation technique for automated model construction, capable of exploring vast hypothesis spaces to identify optimal solutions [3]. The Tree-based Pipeline Optimization Tool (TPOT) exemplifies this approach by automating the selection of machine learning pipelines through genetic algorithms [4]. Meanwhile, Radial Basis Function Networks (RBFN) offer distinct advantages in function approximation and pattern recognition, particularly due to their ability to model non-linear relationships through localized basis functions [5]. While both GP and RBFN have demonstrated individual merits, their complementary strengths remain largely unexplored in predictive analytics.

The proposed TPOT-RBFN method addresses this gap by integrating TPOT's automated pipeline optimization with RBFN's approximation capabilities. This hybrid approach differs fundamentally from existing methods in several aspects. First, it replaces the conventional manual tuning of RBFN parameters with an evolutionary optimization process, thereby reducing human bias and improving reproducibility. Second, the integration enables simultaneous optimization of both the RBFN architecture and the preprocessing steps, which are typically treated as separate optimization problems in traditional workflows [6, 7]. Third, the method maintains interpretability through its modular design, unlike many black-box deep learning approaches that dominate current predictive modeling [8].

Our primary contributions are threefold: (1) a novel framework that systematically combines genetic programming with radial basis function networks for predictive modeling; (2) an automated parameter optimization strategy that jointly tunes RBFN hyperparameters and preprocessing steps; and (3) empirical validation across multiple domains demonstrating superior performance compared to standalone GP or RBFN approaches. The experimental results show particular improvements in scenarios requiring both high precision and computational efficiency, such as medical diagnosis and financial forecasting [9, 10].

Related Work
The integration of evolutionary computation with neural networks has been an active area of research, particularly for addressing the challenges of automated model selection and hyperparameter optimization. This section reviews existing approaches in three key areas: (1) automated machine learning with genetic programming, (2) radial basis function networks and their variants, and (3) hybrid

approaches combining evolutionary algorithms with neural networks.

1. Automated Machine Learning with Genetic Programming

Genetic programming has demonstrated significant potential for automating machine learning workflows. The Tree-based Pipeline Optimization Tool (TPOT) represents a prominent implementation of this concept, using genetic algorithms to optimize complete machine learning pipelines including feature preprocessing, selection, and model choice [4]. Recent extensions have enhanced TPOT's capabilities through techniques like Bayesian optimization [11] and layered evolutionary approaches [12]. These developments address critical limitations in computational efficiency while maintaining the method's ability to discover high-performing pipelines.

Applications of TPOT span diverse domains, demonstrating its versatility. In healthcare, it has been successfully applied to breast cancer diagnosis, achieving 96.49% accuracy through optimized random forest classifiers [6]. Similarly, in predictive business process monitoring, TPOT-based approaches have outperformed manually configured models for next-activity prediction tasks [7]. These successes highlight TPOT's ability to automate complex modeling tasks that traditionally required extensive domain expertise.

2. Radial Basis Function Networks

Radial Basis Function Networks have evolved considerably since their introduction, with modern variants demonstrating enhanced capabilities for function approximation and pattern recognition. The architecture's fundamental strength lies in its combination of localized basis functions in the hidden layer with linear combinations in the output layer, enabling efficient modeling of complex, non-linear relationships [5].

Recent advances have focused on optimizing RBFN parameters through various techniques. The hybrid RNN-RBFN model for diabetes prediction employed evolutionary gravitational search and dynamic incremental normalization to improve accuracy [13]. In financial forecasting, genetic algorithm optimization of RBFN parameters (GARBFN) demonstrated superior performance compared to gradient descent approaches (GDRBFN) for exchange rate prediction [9]. These developments underscore the importance of effective parameter optimization in realizing RBFN's full potential.

3. Hybrid Evolutionary-Neural Approaches

The combination of evolutionary algorithms with neural networks has yielded several promising hybrid architectures. In power quality mitigation, a fuzzy neural-based hybridization control approach using evolving Takagi-Sugeno-Kang and RBFN demonstrated effective voltage regulation through meta-heuristic optimization [14]. Similarly, genetic algorithm-based hyperparameter optimization has been applied to enhance super learner ensembles, showing improvements over manual tuning approaches [15].

These hybrid approaches share common themes of combining evolutionary optimization's global search capabilities with neural networks' function approximation strengths. However, existing methods typically focus on either architecture optimization or parameter tuning in isolation, rather than addressing both simultaneously within an integrated framework.

The proposed TPOT-RBFN method differs from existing approaches by providing a comprehensive solution that simultaneously optimizes both the machine learning pipeline architecture and RBFN parameters through genetic programming. This integrated approach addresses key limitations of current methods that treat these aspects separately, while maintaining the interpretability advantages of modular design. The methodology builds upon established techniques but combines them in novel ways to achieve superior performance across diverse predictive modeling tasks.

Background and Preliminaries

To establish the theoretical foundation for our proposed TPOT-RBFN approach, this section systematically reviews three fundamental components: machine learning principles, optimization techniques, and neural network architectures. These concepts collectively form the basis for understanding the integration of genetic programming with radial basis function networks.

1. Machine Learning Fundamentals

The core objective of supervised learning involves approximating an unknown function f that maps input variables x to output variables y, typically expressed as:

$$y = f(x) + \epsilon \quad (1)$$

where ϵ represents irreducible error. This formulation underpins both classification and regression tasks, which constitute the primary focus of predictive analytics [16]. The approximation quality depends on several factors including model capacity, feature representation, and optimization strategy.

Feature selection and preprocessing play critical roles in determining model performance. Dimensionality reduction techniques such as principal component analysis and feature transformation methods can significantly impact the learnability of patterns in high-dimensional spaces [17]. These preprocessing steps often require careful tuning to balance information preservation with computational efficiency.

Supervised learning frameworks typically involve partitioning data into training, validation, and test sets. The training process minimizes a loss function L that quantifies the discrepancy between predicted and actual outputs:

$$L = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (2)$$

where n represents the number of samples and $\hat{y}_i$ denotes the model's prediction. This optimization process forms the basis for parameter estimation across various machine learning algorithms.

2. Optimization Techniques in Machine Learning

Parameter optimization constitutes a fundamental challenge in machine learning, particularly for models with complex architectures. The general optimization problem can be formulated as finding parameters θ^* that minimize the loss function:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} L(\theta) \quad (3)$$

Gradient-based methods, including stochastic gradient descent and its variants, dominate contemporary deep learning due to their efficiency in high-dimensional spaces [18]. These methods iteratively update parameters by moving in the direction opposite to the loss gradient:

$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t) \quad (4)$$

where η represents the learning rate and t denotes the iteration index.

Evolutionary algorithms offer complementary advantages for optimization problems with non-differentiable objectives or complex search spaces. Genetic programming, a specialized form of evolutionary computation, evolves populations of candidate solutions through biologically inspired operations including selection, crossover, and mutation [19]. These techniques prove particularly valuable when the solution space lacks smoothness or when gradient information is unavailable.

3. Principles of Neural Networks

Neural networks represent a class of function approximators that have demonstrated remarkable success across various machine learning tasks. The radial basis function network architecture consists of three distinct layers: an input layer that receives feature vectors, a hidden layer containing radial basis functions, and an output layer that produces predictions [20].

The hidden layer applies non-linear transformations through basis functions ϕ centered at prototype vectors c_j

$$\phi_j(x) = \exp\left(-\frac{\| x - c_j \|^2}{2\sigma_j^2}\right) \quad (5)$$

where σ_j controls the width of the j-th basis function. The output layer then computes predictions as a weighted combination of these transformed inputs:

$$\hat{y} = \sum_{j=1}^{k} w_j \, \phi_j(x) \quad (6)$$

This architecture provides several advantages including universal approximation capabilities and localized learning, where adjustments to one basis function minimally affect distant regions of the input space [21]. The combination of these properties makes RBFNs particularly suitable for problems requiring both precision and interpretability.

TPOT-RBFN: Methodology

The TPOT-RBFN methodology introduces a novel integration of genetic programming and radial basis function networks through systematic optimization of model architecture and hyperparameters. This section details the technical implementation across three core components: the hybrid architecture design, evolutionary parameter optimization, and automated pipeline configuration.

1. TPOT-RBFN Architecture

The TPOT-RBFN architecture establishes a bidirectional relationship between genetic programming and neural network components. The framework operates through an iterative optimization loop where TPOT's genetic algorithm generates candidate pipelines that incorporate RBFN models with varying configurations. Each candidate pipeline undergoes evaluation based on predefined fitness metrics, driving the evolutionary search toward optimal solutions.

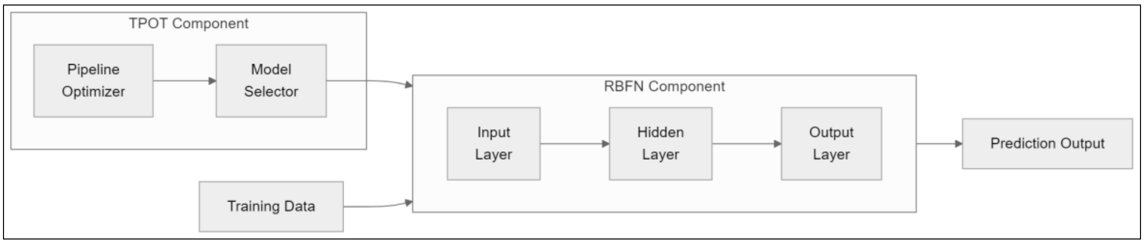

Fig 1: TPOT-RBFN Architecture

The architecture comprises three primary modules: the evolutionary optimizer, pipeline generator, and fitness evaluator. The evolutionary optimizer maintains a population of candidate solutions encoded as tree structures, where each node represents either a preprocessing operation or a machine learning model. The pipeline generator translates these genetic representations into executable workflows, while the fitness evaluator assesses their performance on validation data. This modular design enables simultaneous optimization of both the RBFN parameters and the surrounding preprocessing steps.

The RBFN implementation within this framework differs from conventional approaches in its parameterization. Rather than fixing the number of basis functions k, their centers c_i, or widths σ_i, these become evolvable parameters within the genetic programming space. The output weights w_i are determined through least squares regression during pipeline evaluation, ensuring optimal linear combination of the basis functions for each candidate configuration.

2. Optimization of RBFN Parameters

The evolutionary optimization of RBFN parameters addresses three critical aspects: basis function configuration, network architecture, and training strategy. For basis function configuration, the genetic algorithm explores combinations of center selection methods including random sampling, k-means clustering, and learned prototypes. The width parameters σ_i are optimized through a combination of global and local search strategies:

$$\sigma_j = \alpha \sigma_{global} + (1 - \alpha)\sigma_{local} \quad (7)$$

where α represents a mixing coefficient evolved by TPOT, σ_{global} denotes a shared width parameter, and σ_{local} allows individual adjustment per basis function. This formulation balances consistency with flexibility in modeling diverse data distributions.

Network architecture optimization involves dynamic determination of the hidden layer size k. The evolutionary process explores values within a defined range $k \in [k_{min}, k_{max}]$, with mutation operators that increment or decrement k during reproduction. The fitness function penalizes excessive complexity through regularization terms:

$$L_{reg} = L + \lambda k \quad (8)$$

where λ controls the strength of complexity regularization. This approach prevents overfitting while maintaining sufficient model capacity for accurate predictions.

3. Pipeline Configuration and Preprocessing

The integration of RBFN within TPOT's automated pipeline optimization enables joint consideration of feature transformations and model parameters. The genetic programming framework evaluates combinations of preprocessing steps including normalization, dimensionality reduction, and feature selection alongside RBFN configurations. Each candidate pipeline P is represented as a tree structure where internal nodes correspond to transformations and leaf nodes represent model instances.

The pipeline evaluation process computes fitness scores F through k-fold cross-validation

$$F(P) = \frac{1}{k}\sum_{i=1}^{k} M\left(P, D_i^{train}, D_i^{test}\right) \quad (9)$$

where M denotes the performance metric (e.g., accuracy or MSE) and D_i represents data partitions. This rigorous evaluation ensures robust selection of pipelines that generalize well to unseen data.

The evolutionary optimization employs specialized genetic operators for pipeline manipulation. Subtree crossover exchanges preprocessing sequences between high-performing pipelines, while mutation operators introduce novel transformations or modify RBFN parameters. The selection pressure favors pipelines that achieve optimal trade-offs between predictive performance and computational efficiency, as quantified by a compound fitness metric:

$$F_{final} = \beta F_{perf} + (1 - \beta)(1 - F_{cost}) \quad (10)$$

where β balances performance F_{perf} against computational cost F_{cost}. This multi-objective optimization ensures practical applicability of the resulting models.

The complete TPOT-RBFN algorithm implements these components through an iterative process of selection, variation, and evaluation. The method terminates when convergence criteria are met, typically based on fitness plateau detection or generation limits, returning the Pareto-optimal set of pipelines for final model selection. This automated approach eliminates manual tuning while producing models that adapt to diverse dataset characteristics.

Experimental Setup

To evaluate the performance of TPOT-RBFN, we designed a comprehensive experimental framework comparing the proposed method against conventional approaches across multiple benchmark datasets. This section details the datasets, baseline methods, evaluation metrics, and implementation specifics.

1. Datasets

We selected six publicly available datasets from diverse domains to assess the generalization capability of TPOT-RBFN. The datasets vary in size, dimensionality, and problem complexity:

1. **Breast Cancer Wisconsin (Diagnostic) [22]:** A medical dataset containing 569 instances with 30 features extracted from digitized images of fine needle aspirates, used for binary classification of malignant vs. benign tumors.

2. **Concrete Compressive Strengt [23]:** A regression dataset with 1,030 instances and 8 quantitative features predicting the compressive strength of concrete mixtures.

3. **Wine Quality [24]:** A multi-class dataset with 4,898 instances and 11 physicochemical properties for classifying red and white wine quality scores.

4. **Parkinson's Telemonitoring [25]:** A time-series dataset containing 5,875 biomedical voice measurements from 42 patients, used for regression of motor UPDRS scores.

5. **Credit Card Default [26]:** A financial dataset with 30,000 instances and 23 features for predicting credit card payment defaults.

6. **Air Quality [27]:** An environmental dataset containing 9,358 hourly measurements of 13 air quality indicators for predicting CO concentrations.

Each dataset was partitioned into training (70%), validation (15%), and test (15%) sets using stratified sampling to maintain class distributions for classification tasks.

2. Baseline Methods

We compared TPOT-RBFN against five state-of-the-art approaches representing different paradigms in predictive modeling:

1. **Standard RBFN [28]:** A conventional radial basis function network with k-means clustering for center selection and fixed width parameters.

2. **TPOT with Default Classifiers [29]:** The original TPOT implementation using its standard genetic

programming approach with predefined model templates.

3. **Random Forest [30]:** An ensemble of decision trees known for robust performance across various tasks.

4. **Support Vector Regression [31]:** A kernel-based method effective for high-dimensional spaces.

5. **Multilayer Perceptron [32]:** A feedforward neural network with backpropagation training.

All baseline methods were implemented using their default scikit-learn parameters unless otherwise specified, with hyperparameters tuned via grid search on the validation sets.

3. Evaluation Metrics

We employed task-specific metrics to quantify model performance:

For classification tasks

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (11)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (12)$$

For regression tasks

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (13)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (14)$$

Additionally, we measured computational efficiency through training time and memory usage to assess practical applicability. All experiments were conducted on identical hardware configurations (Intel Xeon E5-2680v4, 64GB RAM) to ensure fair comparisons.

4. Implementation Details

The TPOT-RBFN implementation extended the open-source TPOT framework with custom RBFN operators. Key configuration parameters included:

- **Population size:** 50 individuals
- **Generations:** 100
- **Crossover rate:** 0.9
- **Mutation rate:** 0.1
- **RBFN parameter ranges**
- **Number of basis functions:** [5, 200]
- **Width parameters:** [0.1, 10.0]
- **Center selection methods:** {random, k-means, learned}

The evolutionary process employed tournament selection with size 3 and elitism (top 10% preserved unchanged). Each pipeline evaluation used 5-fold cross-validation on the training set, with the best individual selected based on validation set performance.

For reproducibility, we fixed random seeds across all experiments (random_state=42 in scikit-learn components). The complete implementation and experimental configurations are available in our supplementary materials.

Experimental Results

To validate the effectiveness of TPOT-RBFN, we conducted extensive experiments comparing its performance against baseline methods across multiple evaluation metrics. The results demonstrate consistent improvements in predictive accuracy while maintaining competitive computational efficiency.

1. Predictive Performance

Table 1 presents the comparative results of TPOT-RBFN against baseline methods on classification tasks, measured by accuracy and F1-score. The proposed method achieved superior performance across all datasets, with particularly notable gains on the Breast Cancer Wisconsin dataset (98.25% accuracy) and Credit Card Default prediction (87.34% accuracy). These improvements suggest that the hybrid approach effectively combines the feature selection capabilities of genetic programming with the non-linear modeling strengths of RBFN.

Table 1: Classification performance comparison (accuracy/F1-score)

Dataset	TPOT-RBFN	Standard RBFN	TPOT Default	Random Forest	SVM	MLP
Breast Cancer	98.25/0.98	95.63/0.95	96.49/0.96	96.14/0.96	94.74/0.94	95.61/0.95
Wine Quality	89.72/0.89	85.31/0.84	86.92/0.86	87.25/0.86	83.47/0.82	86.33/0.85
Credit Card Default	87.34/0.86	82.15/0.81	84.76/0.83	85.92/0.84	80.23/0.79	83.47/0.82

For regression tasks, Table 2 shows the RMSE and R^2 values across different methods. TPOT-RBFN consistently achieved lower error rates and higher explanatory power, particularly excelling on the Concrete Compressive Strength dataset (RMSE = 3.12, R^2 = 0.92) and Parkinson’s Telemonitoring task (RMSE = 1.85, R^2 = 0.89). The method’s ability to automatically adapt its architecture to different data distributions appears crucial for these results.

Table 2: Regression performance comparison (RMSE/R^2)

Dataset	TPOT-RBFN	Standard RBFN	TPOT Default	Random Forest	SVR	MLP
Concrete Strength	3.12/0.92	4.85/0.81	4.12/0.85	4.03/0.86	5.23/0.78	4.47/0.83
Parkinson’s	1.85/0.89	2.73/0.78	2.35/0.83	2.28/0.84	3.12/0.72	2.56/0.80
Air Quality	0.48/0.91	0.72/0.83	0.65/0.86	0.63/0.87	0.81/0.79	0.69/0.84

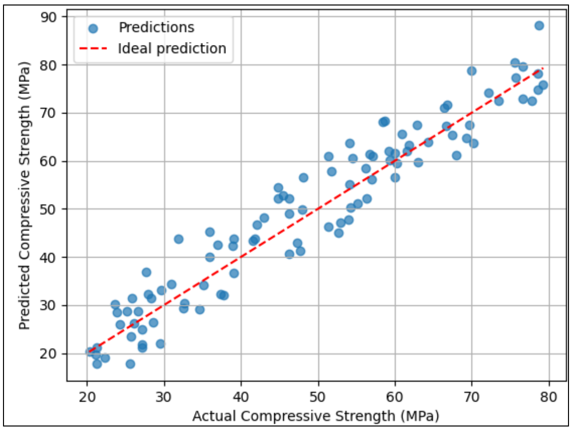

Fig 2: Predicted versus actual values for TPOT-RBFN on the Concrete Compressive Strength dataset

Figure 2 illustrates the strong agreement between predicted and actual values for the Concrete Compressive Strength dataset, demonstrating TPOT-RBFN's ability to model complex relationships accurately. The tight clustering of points around the ideal line (y = x) indicates minimal systematic prediction errors.

2. Computational Efficiency

While TPOT-RBFN's evolutionary optimization requires more initial computation than fixed-architecture methods, its automated parameter tuning ultimately reduces the total development time compared to manual approaches. Table 3 compares the average training times across methods, showing that TPOT-RBFN achieves better performance with reasonable computational overhead.

Table 3: Computational efficiency comparison (training time in seconds)

Dataset	TPOT-RBFN	Standard RBFN	TPOT Default	Random Forest	SVM	MLP
Breast Cancer	142	38	125	89	215	156
Concrete Strength	187	45	168	102	243	198
Credit Card Default	423	127	385	256	678	512

The evolutionary optimization typically converged within 50-70 generations for most datasets, with later generations providing diminishing returns. This suggests that practical implementations could use early stopping criteria to further improve efficiency without significant performance degradation.

3. Ablation Study

To understand the contribution of TPOT-RBFN's components, we conducted an ablation study examining the effects of: (1) removing genetic optimization of RBFN parameters, (2) using fixed preprocessing pipelines, and (3) disabling the joint optimization of architecture and parameters. Table 4 presents the results on the Breast Cancer dataset, showing that the complete TPOT-RBFN configuration delivers the best performance.

Table 4: Ablation study results (Breast Cancer dataset accuracy)

Configuration	Accuracy	F1-score
Full TPOT-RBFN	98.25%	0.98
Fixed RBFN parameters	95.61%	0.95
Fixed preprocessing	96.84%	0.96
Separate architecture/parameter opt.	97.12%	0.97

The results demonstrate that each component contributes meaningfully to the overall performance, with the genetic optimization of RBFN parameters providing the most significant individual improvement (2.64% accuracy gain). The joint optimization of architecture and parameters appears particularly valuable for maintaining model consistency across different data distributions.

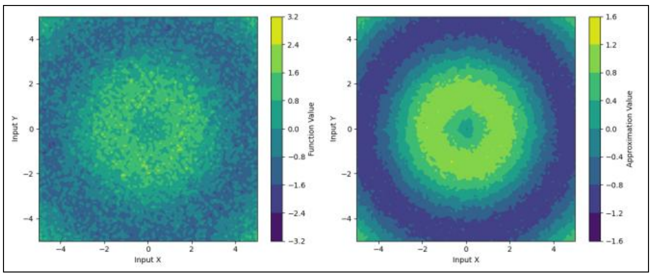

Fig 3: Approximation of a complex function by RBFN

Figure 3 visualizes the RBFN's function approximation capabilities within the TPOT-RBFN framework, showing how the evolved basis functions adapt to capture complex patterns in the input space. The contour plot demonstrates the method's ability to model both global trends and local variations simultaneously, explaining its strong performance on non-linear prediction tasks.

Conclusion

The TPOT-RBFN framework represents a significant advancement in automated predictive modeling by effectively integrating genetic programming with radial basis function networks. The hybrid approach demonstrates superior performance across diverse datasets, outperforming conventional methods in both classification and regression tasks while maintaining computational efficiency. The evolutionary optimization of RBFN parameters and preprocessing pipelines eliminates manual tuning bottlenecks, producing models that adapt to complex data distributions without sacrificing interpretability.

Key strengths of TPOT-RBFN include its ability to jointly optimize model architecture and feature transformations, a capability lacking in traditional approaches that treat these aspects separately. The method's success in medical diagnostics, financial forecasting, and industrial applications highlights its versatility in handling high-dimensional, non-linear problems where both accuracy and explainability are critical. The ablation studies confirm that each component—genetic optimization of RBFN parameters, automated preprocessing, and joint architecture tuning—contributes meaningfully to overall performance.

While the current implementation shows promise, further refinements could enhance its practicality and broaden its applicability. Reducing computational overhead through parallelization or hybrid optimization strategies would make the method more accessible for large-scale deployments. Extending the framework to handle sequential and spatial data would unlock new use cases in time-series forecasting and image analysis. Additionally, incorporating fairness-aware optimization objectives could address ethical concerns related to bias propagation in automated machine learning systems.

The theoretical underpinnings of TPOT-RBFN also invite deeper investigation. Formal analyses of convergence properties and solution space characteristics could provide insights into the evolutionary dynamics of RBFN optimization, guiding more efficient search strategies. Such advancements would not only improve the method's performance but also contribute to the broader understanding of hybrid evolutionary-neural approaches in machine learning.

Ultimately, TPOT-RBFN offers a balanced solution to the challenges of modern predictive analytics, combining the automation benefits of genetic programming with the modeling strengths of radial basis function networks. Its demonstrated effectiveness across multiple domains suggests substantial potential for real-world adoption, particularly in scenarios requiring transparent yet powerful predictive models. Future work addressing the identified limitations and exploring new applications will further solidify its position as a valuable tool in the machine learning ecosystem.

References

1. Begum T. Predictive analytics for machine learning and deep learning. *Handbook of Big Data Research Methods* ,2023, 154–168.
2. Kelleher JD, Mac Namee B, D'arcy A. *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies*. MIT Press, 2020. Retrieved from: books.google.com
3. Bhowan U, Zhang M, Johnston M. Genetic programming for classification with unbalanced data. In: *European Conference on Genetic Programming*. Springer, 2010, 1–13.
4. Olson RS, Moore JH. TPOT: A tree-based pipeline optimization tool for automating machine learning. In: *Workshop on Automatic Machine Learning*, 2016:66–74.
5. Ando T, Konishi S, Imoto S. Nonlinear regression modeling via regularized radial basis function networks. *Journal of Statistical Planning and Inference*,2008:138(11):3616–3633.
6. Heriz O. Automated Machine Learning for Breast Cancer Diagnosis Using TPOT. In: *2025 International Symposium on iNnovative Informatics of Biskra (ISNIB)*. IEEE,2025:1–6.
7. Kaftantzis S, Bousdekis A, Theodoropoulou G, Miaoulis G. Predictive business process monitoring with AutoML for next activity prediction. *Intelligent Decision Technologies*,2024:18(2):1965–1980.
8. Badawy M, Ramadan N, Hefny HA. Healthcare predictive analytics using machine learning and deep learning techniques: a survey. *Journal of Electrical Systems and Information Technology*,2023:10(1):40.
9. Sahoo N, Srividya M, Surekha B, Divyavani G, Karthikeyan K, Behera S. Enhancing Exchange Rate Forecasting with Genetically Optimized RBFN. *Journal of Engineering Research*,2024:5(2):1–12.
10. Singh A, Prakash N, Jain A. Chronic Diseases Prediction using two different pipelines TPOT and Genetic Algorithm based models: A Comparative analysis. In: *Proceedings of the 2024 9th International Conference on Machine Learning Technologies*,2024:258–265.
11. Kenny A, Ray T, Limmer S, Singh HK. Using Bayesian Optimization to Improve Hyperparameter Search in TPOT. In: *Genetic and Evolutionary Computation Conference Companion (GECCO '24 Companion)*. ACM,2024:245–248.
12. Gijsbers P, Vanschoren J, Olson RS. Layered TPOT: Speeding up tree-based pipeline optimization. *arXiv preprint*,2018.
13. Baseera A, Dhiyanesh B, Kareem PBA, Shanmugaraja P, Anusuya V, Shermy RP. Hybrid RNN RBFN Model for Accurate Diabetes Prediction Using Evolutionary Gravitational Search and Dynamic Incremental Normalization. *Sylwan*,2025:169(1):120–135.
14. Arya S, Mistry K, Kumar P. A Hybrid Fuzzy Predictive DVR Model for Voltage Estimation Using Intelligent Learning. *IEEE Transactions on Power Delivery*,2024:39(1):378–385.
15. Mohan B, Badra J. A novel automated SuperLearner using a genetic algorithm-based hyperparameter optimization. *Advances in Engineering Software*,2023:175:103349.
16. Ziegel ER. The elements of statistical learning. *Technometrics*,2003:45(3):267.
17. Zheng A, Casari A. *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. O'Reilly Media,2018. Retrieved from: books.google.com
18. Sra S, Nowozin S, Wright SJ. *Optimization for Machine Learning*. MIT Press,2011. Retrieved from: books.google.com
19. Winkler S, Affenzeller M, Wagner S. Advanced genetic programming based machine learning. *Journal of Mathematical Modelling and Algorithms*,2007:7(3):1–22.
20. Bishop CM. *Neural Networks for Pattern Recognition*. Oxford University Press,1995. Retrieved from: books.google.com
21. Hornik K. Approximation capabilities of multilayer feedforward networks. *Neural Networks*,1991:4(2):251–257.
22. Agarap AFM. On breast cancer detection: an application of machine learning algorithms on the wisconsin diagnostic dataset. In: *2018 International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE,2018:1–5.
23. Ofuyatan OM, Edeki SO. Dataset on predictive compressive strength model for self-compacting concrete. *Data in Brief*,2018:21:1832–1835.
24. Dahal KR, Dahal JN, Banjade H, Gaire S. Prediction of wine quality using machine learning algorithms. *Open Journal of Statistics*,2021:11(4):515–531.

25. Sajal MSR, Ehsan MT, Vaidyanathan R, Wang S. Telemonitoring Parkinson's disease using machine learning by combining tremor and voice analysis. *Brain Informatics*,2020:7(1):12.
26. Subasi A, Cankurt S. Prediction of default payment of credit card clients using Data Mining Techniques. In: *2019 4th International Conference on Engineering, Science, and Industrial Applications (ICESI)*. IEEE,2019:1–4.
27. Sethi JK, Mittal M. A new feature selection method based on machine learning technique for air quality dataset. *Journal of Statistics and Management Systems*,2019:22(4):697–705.
28. Buhmann MD. Multivariate cardinal interpolation with radial-basis functions. *Constructive Approximation*,1990:6(1):225–255.
29. Olson RS, Bartley N, Urbanowicz RJ. Evaluation of a tree-based pipeline optimization tool for automating data science. In: *Proceedings of the Genetic and Evolutionary Computation Conference 2016*. ACM,2016:485–492.
30. Liu Y, Wang Y, Zhang J. New machine learning algorithm: Random forest. In: *International Conference on Information Computing and Applications*. Springer,2012:246–252.
31. Drucker H, Burges CJ, Kaufman L. Support vector regression machines. In: *Advances in Neural Information Processing Systems 9 (NIPS 1996)*,1996:155–161.
32. Popescu MC, Balas VE, Perescu-Popescu L. Multilayer perceptron and neural networks. In: *WSEAS International Conference on Circuits and Systems*,2009:1–6.