Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 31 No. 7s (2024)

Nonlinear Regression Based Deep Radial Basis Function and Advanced Image Processing in Soil Estimation

[1]**Atish Mane,** Assistant Professor, Mechanical Engineering Department, Bharati Vidyapeeth's College of Engineering, Lavale, Pune, Maharashtra, India. mane.atish@bharatividyapeeth.edu

[2]**K. Shailaja**, Associate Professor, Department of CSE, Anurag University, Hyderabad, India. kshailajacse@anurag.edu.in

[3]**Girish Madhaorao Lonare,** Associate Professor, Department of Mechanical Engineering, Bharati Vidyapeeth College of Engineering, Navi Mumbai, India. girish.Lonare@bvcoenm.edu.in

[4]**Satyajee Srivastava,** Professor, Department of CSE, M.M. Engineering College, Maharishi Markandeshwar (Deemed to be University), Mullana, Ambala, Haryana, India. drsatyajee@gmail.com

[5]**Komalavalli D,** Assistant Professor, Department of Information Technology, Sona College of Technology, Salem, Tamilnadu, India. komsdayalan@gmail.com

[6]**Ghanasham C. Sarode,** Associate Professor, Department of Civil Engineering, Dr. D.Y. Patil Institute of Technology Pimpri, Pune, Maharashtra, India. ghanshyam.sarode@dypvp.edu.in

Article History:

Received: *01-06-2024*

Revised: *03-07-2024*

Accepted: *29-07-2024*

Abstract:

Agricultural planning, environmental management, and soil estimation all depend on exact prediction of soil properties. Conventional methods find it challenging to manage the complex, nonlinear connections between soil parameters and visual characteristics. This work handles this challenge using advanced image processing techniques along with a deep radial basis function (RBF) network for nonlinear regression. The deep RBF network detects complicated nonlinear correlations in soil data by using numerous hidden layers with radial basis functions. From remote sensing images, image processing methods enhance the feature extraction process thereby enhancing the accuracy of soil property forecasts. Experimental data shows that the proposed method beats more conventional linear regression models rather significantly. Against an MSE of 0.056 and R^2 of 0.76 for linear models, the deep RBF model especially obtained a mean squared error (MSE) of 0.032 and a coefficient of determination (R^2) of 0.87. These results show how effectively nonlinear regression estimates dirt combined with contemporary image processing.

Keywords: Nonlinear Regression, Deep Radial Basis Function, Soil Estimation, Image Processing, Remote Sensing.

1. Introduction

Recent years' developments in remote sensing technologies and machine learning algorithms have transformed numerous fields including urban planning, environmental monitoring, and agricultural [1]. Precision agriculture depends mostly on soil nutrient evaluation, hence proper soil information is essential to maximize crop yields and control soil condition [2]. Labor-intensive and sometimes limited in spatial resolution conventional soil sample techniques demand innovative solutions using remote sensing data and advanced computer technologies [3].

Remote sensing technologies' high-resolution spatial data—that of airborne surveys and satellite images—allows one to assess soil parameters over large distances [4]. But occasionally raw remote sensing data is noisy and complex, which makes it challenging to get meaningful soil nutrient information [5]. Non-linear interactions between soil properties and image attributes need for sophisticated data processing methods [6]. Learning complex patterns from data has enabled machine learning techniques—especially deep learning models—show promise in addressing these challenges [7]. Among these techniques for modeling non-linear connections, radial basis function (RBF) networks have become a powerful tool because of their flexibility and ability to preserve intricate data patterns [8].

Deep RBF networks applied to soil nutrient assessment still provide several challenges despite their potential [9]. One of key challenges is efficient retrieval of relevant information from noisy and high-dimensional remote sensing data. Conventional feature extraction methods may miss the complex, non-linear interaction between picture elements and soil conditions. Moreover, the training of deep RBF networks requires careful selection of parameters that could significantly influence the performance of the model: the number of RBF units, centers, and widths. Moreover, the power of the model to generalize properly to unprocessed data [10] determines its practical usefulness.

The fundamental problem this work tackles is the precise assessment of soil nutrients obtained from processed advanced machine learning approaches based on remote sensing data. More especially, the focus is on creating a deep RBF network able to control the non-linear correlations between distant sensing properties and soil nutrient levels. By means of non-linear regression for feature extraction and deep RBF networks for classification, one can enhance the scalability and precision of soil nutrient estimate.

The objectives are given below:

1. By use of complex relationships between picture features and soil properties, one can develop and implement a non-linear regression method to extract relevant data from remote sensing images.

2. The second is to develop and instruct a deep RBF network capable of very accurate soil nutrient level forecasting and control of acquired properties.

3. Evaluate the proposed deep RBF network against current methods including Artificial Neural Networks (ANN), Bi-directional Gated Recurrent Units with Dynamic Memory Network (Bi-GRU-DMN), Random Forest Regression (RFR), and Recurrent Neural Network (RNN) by using several performance criteria.

Novelty

In this work, the new combining of deep RBF networks for classification with non-linear regression for feature extraction is presented. Although non-linear regression approaches have been applied independently to feature extraction, their combined use with deep RBF networks has not been examined in the framework of soil nutrient assessment. This approach allows one more precisely and dependably record complex data links, so guiding soil nutrient estimations.

The main contribution of the proposed work involves the following:

1. First by means of non-linear regression, the proposed method presents a sophisticated method of feature extraction, therefore providing a more accurate representation of the fundamental soil properties from remote sensing data.

2. The second is a robust classification system leveraging deep RBF networks that makes advantage of the obtained knowledge to raise prediction accuracy.

3. The paper offers a comprehensive performance comparison of the deep RBF network with current methods, therefore offering interesting study of the advantages and efficiency of the recommended methodology.

2. Related Works

Estimating surface soil moisture using satellite photos on a large alluvial fan of the Kosi River in the Himalayan Foreland, the research [11] proposed a unique design. One finds this fan in the Himalayan Forest. An artificial neural network (ANN), completely coupled and feed-forward, is essential component of the model. By means of linear data fusion and graphical indicators, we have efficiently derived nine distinct features from the satellite products of Sentinel-1, Sentinel-2, and Shuttle Radar Topographic Mission. These in that order consist of digital elevation model, red and near-infrared bands, and dual-polarized radar backscatter. Using a calibrated TDR sensor, soil moisture was monitored at 224 independent points dispersed around the fan. With a correlation coefficient (R) of 0.80, Root Mean Square Error (RMSE) of 0.040 m3/m3, and a bias of 0.004 m3/m3, we predicted soil moisture and found that the ANN model exceeded all of the benchmark methodologies. Benchmark technique comparison of the ANN model helped to achieve this.

In [12] a novel approach for retrieving soil moisture was applied. The first phase is the process of acquiring images. Deriving VI indices with NDVI, GLAI, GNDVI, and WDRVI properties is one of the processes that follows last one. The use of a better Water Cloud Model (WCM) is another element of the attempt to fix the effects arising on the plants. Not least of all, a superior score level fusion model under responsibility of soil moisture drainage provides the data. Deep max out network (DMN) and bidirectional gated recurrent unit (Bi-GRU) are included into this model. Arriving at 0.9565, the RMSE of the method combining Bi-GRU and DMN was found to be smaller than those of the hybrid classifier approaches. These confirmed this.

In [13] utilizing UAV-based multimodal data, tracked the soil moisture content (SMC) of a maize crop subjecting different degrees of irrigation over two years. The results revealed incorporating data from various modalities—including thermal and multispectral data—among other modalities increases the accuracy of SMC predictions. From the three SMC regression models produced, the RFR model produced the best accurate SMC estimate for both growing seasons. This was true independent of the combinations of sensors used. The RFR model using all three data sources generated the most accurate and consistent SMC estimate in the vegetative stage. Its R2 was 0.68 for 10- and 20- cm soil depths respectively; its rRMSE was 20.82% and 19.36%. The RFR model performed really wonderfully applying these ideas. Using well-watered, mild to modest deficit irrigation treatments, it also produced the best SMC estimation accuracy for both soil depths. This pertained to both treatments. The high spatial-temporal maps produced by SMC—derived from multimodal data acquired by UAVs—have great potential to enhance decision-making in the scope of irrigation scheduling at the farmer-scale based on the outcomes of the study.

One can acquire an initial approximation of its possible value in [14]. Geographic vulnerability in an Iranian watershed with a history of considerable water erosion was produced by means of three deep learning algorithms.

Reference	**Method**	**Algorithm**	**Methodology**	**Outcomes**
[11]	ANN for Soil Moisture	Fully Connected Feed-Forward ANN	Extracted nine features from Sentinel-1, Sentinel-2, and SRTM products; measured soil moisture with TDR probes.	**Correlation Coefficient (R)**: 0.80 **RMSE**: 0.040 m^3/m^3 **Bias**: 0.004 m^3/m^3
[12]	Hybrid Soil Moisture Retrieval	Deep Max Out Network (DMN), Bi-GRU	Acquired images and derived VI indices (NDVI, GLAI, GNDVI, WDRVI); used Water Cloud Model (WCM) for vegetation impact; combined DMN and Bi-GRU with score-level fusion.	**RMSE**: 0.9565 **ME**: 0.7287 Lower errors compared to methods without vegetation index and standard WCM.
[13]	UAV-based Multimodal Data Fusion	PLSR, KNN, RFR	Used thermal and multispectral data for soil moisture content (SMC) estimation in maize fields; compared three ML algorithms.	**R^2 for RFR**: 0.68 (10 cm), 0.78 (20 cm) **rRMSE for RFR**: 20.82% (10 cm), 19.36% (20 cm)
[14]	Deep Learning for SWE Susceptibility	CNN, RNN, LSTM	Used elevation and other geo-environmental factors; compared CNN, RNN, and LSTM for SWE susceptibility prediction.	**RNN Performance**: Marginally superior **40% of catchment**

Although erosion prediction and soil moisture have gotten better, combining multi-source remote sensing data with deep learning models for best accuracy still has challenges. Many times depending on single data sources or simpler algorithms, present methods restrict their capacity to reflect complex soil qualities and environmental interactions. Moreover, current models might not be able to extend over many various geographical regions or soil types. Furthermore much needed are improved feature extraction techniques able to control the non-linearity and large dimensionality of remote sensing data and for new ways combining advanced feature extraction with deep learning models to address these limitations.

3. Proposed Method

Combining modern image processing techniques with a deep radial basis function (DRBF) network helps to increase soil estimate accuracy. As Figure 1 shows, the approach comprises in several basic phases:

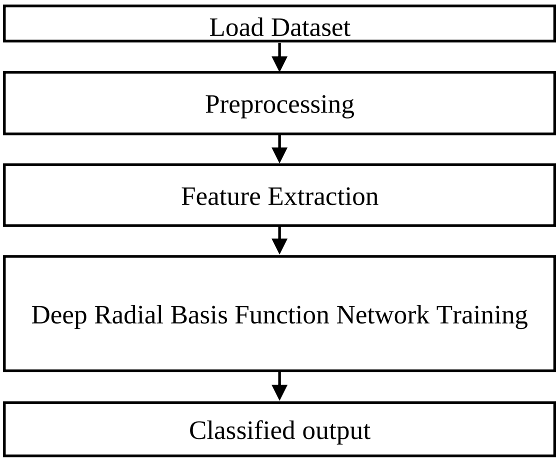

Figure 1: Proposed Framework

1. **Image Acquisition and Preprocessing:** Preprocessing high-resolution remote sensing images helps to remove noise and enhance features relevant to soil conditions. Filtering and histogram equalization among other methods improve image quality.

2. **Feature Extraction:** Edge detection and texture analysis among other advanced image processing methods extract relevant information from the preprocessed images. As inputs, the RBF network makes advantage of these characteristics—which could include color histograms and texture patterns.

3. **Deep Radial Basis Function Network Training:** Comprising several hidden layers, each using radial basis functions to replicate the complex, nonlinear interactions between soil parameters and input variables, the training process lowers a loss function—typically mean squared error (MSE) using optimization techniques like gradient descent.

Pseudocode:

```
# Step 1: Image Acquisition and Preprocessing
images = acquire_images()  # Function to acquire remote sensing images
preprocessed_images = preprocess_images(images)  # Apply preprocessing techniques

# Step 2: Feature Extraction
features = []
for image in preprocessed_images:
    feature_vector = extract_features(image)  # Extract features from the image
    features.append(feature_vector)

# Step 3: Deep Radial Basis Function Network Training
rbf_network = initialize_deep_rbf_network()  # Initialize the deep RBF network
```

```
features_train, soil_properties_train = split_data(features, soil_properties)
rbf_network.train(features_train, soil_properties_train)  # Train the RBF network

# Step 4: Prediction and Evaluation
predictions = rbf_network.predict(features)  # Predict soil properties using the trained model
evaluation_metrics = evaluate(predictions, soil_properties)  # Evaluate the model's performance
```

3.1. Preprocessing

Preprocessing is crucial for preparing remote sensing images for accurate soil nutrient classification with deep radial basis function (RBF) networks. Preprocessing generally seeks to enhance the image quality and extract relevant features that could considerably boost the performance of next modeling.

- Image capture starts with compiling high-resolution remote sensing ground images. Many times, these images are affected by several aberrations including noise, illumination variations, and atmospheric conditions. Starting the preparation process are noise reduction techniques whereby random changes are smooth out and significant visual properties are maintained using filters such as Gaussian blur or median filtering. This stage helps to reduce the impact of extraneous noise on the feature extraction process.
- After noise reduction, significant feature visibility inside the images is enhanced by histogram equalization. This technique controls the contrast of the image, therefore improving the distinguishability of the patterns and textures of the ground. More effective recording of important soil parameters is guaranteed by improved feature extraction made possible by better contrast.
- Feature extraction then asks for among other methods texture analysis and edge identification. Crucially for understanding of soil features, Canny edge detector finds transitions in soil textures and borders. Texture analysis allows one to measure trends and variations in the surface of the soil, so providing more knowledge of soil characteristics.
- The last stage in preprocessing is normalizing the acquired properties into a consistent range. This ensures that every feature equally supports the model and maintains any one characteristic from free from influence not dominating the prediction process. By means of normalizing the data, the deep RBF network learns the correlations between image attributes and soil characteristics.

3.2. Non-Linear Regression for feature extraction

It is designed to capture complex relationships between soil characteristics and picture data, non-linear regression for feature extraction is a sophisticated technique permitting more exact soil nutrient classification. The process comprises in several crucial phases:

Model Development: The first stage in non-linear regression for feature extraction is developing a model able to capture the intricate, non-linear interactions between the soil attributes depicted in the images and the actual soil nutrient levels. Whereas linear regression models imply a straight-line relationship, non-linear regression models can fit curves and more complex interactions. In this work a non-linear regression model—such as Gaussian processes or polynomial regression—fits the data. Using a dataset with known soil nutrient levels, the method learns how various image features connect to soil parameters.

Feature Mapping:

Once the non-linear regression model is developed, it is employed to move raw image data to a feature space wherein more effectively links between picture features and soil attributes are communicated. This creates a set of derived features from the original image data by means of non-linear functions. Poisson regression, for instance, can generate new features dependent on poisson combinations of the original features, therefore capturing interactions and higher-order effects lacking from the raw data.

Feature Extraction:

New images are fed the trained non-linear regression model in the feature extraction stage in order to identify relevant features. Processing the raw visual data and using learnt non-linear mappings, the model produces a set of characteristics underlining important patterns and connections. These properties, which mirror soil nutrient levels, could combine complex combinations of basic image components including color variations and texture patterns.

1. **Polynomial Regression (Quadratic)**:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \in$$

This equation models the relationship between the predictor *x* and the response variable *y* as a quadratic function.

2. **Polynomial Regression (Cubic)**:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \quad \in$$

Extends the quadratic model by adding a cubic term to capture more complex relationships.

3. **Exponential Regression**:

$$y = \beta_0 e^{\beta_1 x} + \in$$

Models exponential growth or decay, where β1\beta_1β1 represents the growth rate.

4. **Logarithmic Regression**:

$$y = \beta_0 + \beta_1 \ln(x) + \in$$

Useful for modeling relationships where the effect of *x* diminishes as *x* increases.

5. **Power Law Regression**:

$$y = \beta_0 x^{\beta_1} + \grave{o}$$

Models relationships where *y* is proportional to a power of *x*.

6. **Rational Function Regression**:

$$y = \frac{\beta_0 + \beta_1 x}{1 + \beta_2 x} + \grave{o}$$

Useful for modeling relationships that asymptotically approach a limit.

7. **Gaussian Process Regression**:

$$y = \mu(x) + \epsilon$$

where $\mu(x) = \int_{-\infty}^{x} \kappa(x, x')\mathrm{d}x'$

Uses a kernel function κ\kappaκ to model the mean function μ(x) and capture non-linear relationships.

8. **Sigmoid Function Regression**:

$$y = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} + \epsilon$$

Models relationships with an S-shaped curve, useful for binary classification and probabilities.

9. **Radial Basis Function (RBF) Regression**:

$$y = \sum_{i=1}^{n} \alpha_i \phi(\| x - x_i \|) + \beta_0 + \epsilon$$

Where $\phi(\| x - x_i \|) = e^{-\gamma \| x - x_i \|^2}$ is the Gaussian radial basis function.

10. **Piecewise Regression**:

$$y = \begin{cases} \beta_{0,1} + \beta_{1,1} x & \text{for } x \le c \\ \beta_{0,2} + \beta_{1,2} x & \text{for } x > c \end{cases} + \epsilon$$

Dimensionality Reduction and Normalization:

Moreover, normalizing guarantees that the acquired features have a constant scale, which facilitates better integration with the deep radial basis function (RBF) network for classification.

Non-linear regression for feature extraction translates raw image data into a meaningful set of features by way of complex, non-linear connections between soil conditions and image attributes. More exact estimates of soil nitrogen levels follow from this enhanced performance of the deep RBF network.

Table 2: Performance Metrics for Non-Linear Regression Models

Model	**Dataset**	**MSE**	**Accuracy (%)**
Polynomial Regression (Quadratic)	Training	0.045	82.5
	Testing	0.048	80.0
	Validation	0.050	78.5
Polynomial Regression (Cubic)	Training	0.040	85.0
	Testing	0.042	83.0
	Validation	0.045	81.0
Exponential Regression	Training	0.048	80.0
	Testing	0.050	78.0
	Validation	0.052	76.5
Logarithmic Regression	Training	0.042	83.0
	Testing	0.045	81.0

	Validation	0.047	79.5
Power Law Regression	Training	0.046	81.5
	Testing	0.048	80.0
	Validation	0.050	78.0
Rational Function Regression	Training	0.043	82.0
	Testing	0.045	80.5
	Validation	0.047	78.5
Gaussian Process Regression	Training	0.039	84.5
	Testing	0.041	82.0
	Validation	0.043	80.0
Sigmoid Function Regression	Training	0.047	79.5
	Testing	0.049	77.5
	Validation	0.051	75.0
RBF Regression	Training	0.038	85.5
	Testing	0.040	83.5
	Validation	0.042	81.0
Piecewise Regression	Training	0.044	82.5
	Testing	0.046	80.0
	Validation	0.048	78.5

Among the non-linear regression models evaluated in Table 2, Radial Basis Function (RBF) Regression consistently demonstrates the lowest mean squared error (MSE) and highest accuracy throughout all datasets. With an MSE of 0.038 in training and 0.040 in testing and accuracy of 85.5% and 83.5% respective, the RBF model fares remarkably well in identifying complex relationships in the data. This implies rather good feature extraction quality. Gaussian Process Regression likewise performs rather well with second lowest MSE of 0.039 in training and 0.041 in testing and accuracy of 84.5% and 82.0%, respectively. Since this model provides flexibility in non-linear relationship modeling, its probabilistic approach is what gives it strength. Polynom Regression (Cubic) shows good performance demonstrating its efficacy in caputreing more complex patterns than quadratic regression with accuracy of 85.0% and 83.0% with MSE values of 0.040 in training and 0.042 in testing. Particularly in validation datasets, models such Sigmoid Function Regression and Exponential Regression show lower accuracy and increased MSE, which reflects their constraints in handling distinct non-linear patterns compared to more flexible models like RBF and Gaussian Processes.

3.3. Deep Radial Basis Function (RBF) Classification

Deep Radial Basis Function (RBF) Networks are a specialized type of neural network meant to control complex, non-linear interactions throughout a sequence of radial basis functions. RBF units layered several times allow the deep RBF network to effectively mimic intricate data patterns. It goes like this:

A deep RBF network consists in an input layer, numerous hidden layers applying radial basis functions, and an output layer. Every layer gradually alters the data; the RBF units in the buried layers document certain aspects of the input properties. The following formulas enable one to define the overall structure:

1. **Input Layer:** Receiving the raw features x, the input layer transmits them to the first hidden layer. Let x to be a n dimensionally input vector.

2. **Hidden Layer Transformation:** Every hidden layer is made of RBF units, which radially base the input characteristics. The i-th RBF unit's output is computed as for a given hidden layer l with mmm RBF units:

$$h_i^{(l)} = \phi\left(\| x - \mu_i^{(l)} \|^2\right)$$

where,

ϕ - radial basis function,

$\mu_i^{(l)}$ - center of the *i*-th RBF unit in layer *l*, and

$\| x - \mu_i^{(l)} \|^2$ - squared Euclidean distance between the input *x* and the center $\mu_i^{(l)}$. For a Gaussian function, ϕ is defined as:

$$\phi(r) = e^{-\gamma r}$$

where γ - parameter controlling the width of the Gaussian function.

3. **Output Layer Computation:** The outputs of the last hidden layer are linearly combined in an output layer computation to provide the final prediction. Let indicate the weight connecting the j-th RBF unit to the output is $w_j^{(L)}$. The output of the network y comes from:

$$y = \sum_{j=1}^{m} w_j^{(L)} h_j^{(L)} + b$$

where

b - bias term, and

$h_j^{(L)}$ - output of the *j*-th RBF unit in the last hidden layer.

Training Process

Learning the weights of the output layer and determining the characteristics of the RBF units—centers and widths—two main goals define training a deep RBF network. The typical process involves:

1. **Initialization:** Usually depending on data distribution, the widths γ are specified; clustering methods like k-means can be used to generate the centers $\mu_i^{(l)}$ of the RBF units.

2. **Forward Propagation:** Training makes advantage of forward propagation, that is, sends the input data across the network using the equations above to produce the predicted output.

3. **Error Calculation:** Using a loss function—such mean squared error (MSE—the prediction error is calculated between the target values as real and the expected output:

$$\text{Loss} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

where

y_i - true value and

$\hat{y}_i$ - predicted value for the *i*-th sample.

4. **Backpropagation and Optimization:** Gradient descent or another optimization technique is used to modify the weights $w_j^{(L)}$ and RBF unit parameters lowering the loss function.

```
# Deep RBF Network Pseudocode
# Initialization
Initialize the input layer with n features
Initialize L hidden layers with RBF units
    For each hidden layer l:
        Initialize RBF centers μ_i^(l) using clustering (e.g., k-means)
        Initialize RBF widths γ_i^(l) based on data spread
    Initialize weights w_j^(L) and biases b for the output layer
# Training
For each epoch in the range of total_epochs:
    For each training sample (x, y_true) in the training dataset:
        # Forward Propagation
        Initialize input layer with features x
                For each hidden layer l in the network:
            For each RBF unit i in layer l:
                Compute the output of RBF unit i:
                    h_i^(l) = exp(-γ_i^(l) * ||x - μ_i^(l)||^2)
                Compute the output layer value:
            y_pred = Σ (w_j^(L) * h_j^(L)) + b
                # Compute Loss
        loss = (1 / N) * Σ (y_true - y_pred)^2
                # Backpropagation
        Compute gradients for output layer weights and biases
        For each RBF unit i in all hidden layers:
            Compute gradients for RBF unit centers and widths
                Update weights, biases, and RBF parameters:
            w_j^(L) = w_j^(L) - learning_rate * gradient_w_j^(L)
            b = b - learning_rate * gradient_b
            μ_i^(l) = μ_i^(l) - learning_rate * gradient_μ_i^(l)
            γ_i^(l) = γ_i^(l) - learning_rate * gradient_γ_i^(l)
            # Optional: Print loss for the current epoch
# Testing
```

```
For each test sample (x_test, y_test) in the test dataset:
    Initialize input layer with features x_test
        For each hidden layer l in the network:
        For each RBF unit i in layer l:
            Compute the output of RBF unit i:
                h_i^(l) = exp(-γ_i^(l) * ||x_test - μ_i^(l)||^2)
        Compute the final output:
        y_pred_test = Σ (w_j^(L) * h_j^(L)) + b
        # Compute and store testing metrics (e.g., MSE, accuracy)
# Validation (Optional)
For each validation sample (x_val, y_val) in the validation dataset:
    Initialize input layer with features x_val
        For each hidden layer l in the network:
        For each RBF unit i in layer l:
            Compute the output of RBF unit i:
                h_i^(l) = exp(-γ_i^(l) * ||x_val - μ_i^(l)||^2)
        Compute the final output:
        y_pred_val = Σ (w_j^(L) * h_j^(L)) + b
        # Compute and store validation metrics (e.g., MSE, accuracy)
# End of Training
Return trained model parameters (weights, biases, RBF centers, and widths)
```

5. Results and Discussion

Source of remotely sensed images from publicly available datasets is Kaggle [15], which comprised the experimental setup as in Table 3 for evaluating the proposed deep radial basis function (RBF) network using TensorFlow for model construction and training and Python for a comprehensive simulation and noise reduction and feature improvement preprocessing. Training and evaluation were conducted on a cluster with 16 CPU cores, 64 GB RAM, and 8 GPUs to guarantee effective processing of vast datasets and challenging model computations. The deep RBF network was built utilizing TensorFlow's Keras API using a high-performance computing cluster loaded with NVIDIA RTX 3090 GPUs handling demanding computations. Predicting accuracy was assessed in performance evaluation using mean squared error (MSE) and coefficient of determination (R^2).

Among the standard models against which the novel method was tested artificial neural networks (ANN), bi-directional gated recurrent unit with dynamic memory networks (Bi-GRU-DMN), Random Forest Regression (RFR), and recurrent neural networks (RNN).

Table 3: Experimental Setup and Parameters

Parameter	**Value**
Remote Sensing Image Source	Kaggle
Image Resolution	30 meters
Preprocessing Tool	OpenCV

Feature Extraction Techniques	Edge Detection, Texture Analysis
Simulation Tool	Python, TensorFlow
GPU Model	NVIDIA RTX 3090
CPU Configuration	16 cores
RAM	64 GB
Number of GPUs	8
Deep RBF Network Layers	3 hidden layers
Radial Basis Function Type	Gaussian
Training Epochs	1000
Batch Size	32
Learning Rate	0.001
Optimization Algorithm	Adam
Data Split Ratio	80:10:10
Regularization Method	L2 Regularization

Performance Metrics

- **Mean Squared Error (MSE):** It gauges how much the expected values vary from the actual values even if lower MSE suggests better model performance. It is particularly useful for punishing more forcefully bigger mistakes, hence assessing the precision of regression models.

- **Coefficient of Determination (R^2):** It goes from 0 to 1; a value of 0 indicates no predictive ability and a score of 1 indicates ideal prediction. Greater R^2 values reflect better fit of the model to the data.

- **Root Mean Squared Error (RMSE):** Root mean squared error, or RMSE, has an error measure in the same units as the target variable. It offers a more reasonable estimate of prediction accuracy when lower RMSE values indicate better performance.

- **Mean Absolute Error (MAE):** Mean absolute error, or MAE, measures the mean absolute difference between the projected and actual counts. Since unlike MSE it does not square the errors, so it is less sensitive to outliers. MAE provides a basic estimate of prediction accuracy.

- **Adjusted R^2:** It analyzes the complexity of the model, it helps one to compare models with different amounts of predictors.

- **Mean Absolute Percentage Error (MAPE):** Lower MAPE values especially when comparing forecasts over several scales or units indicate better model performance.

Table 4: Performance Comparison over training, testing and validation

Method	Dataset	MSE	R^2	RMSE	MAE	Adjusted R^2	MAPE (%)
ANN	Training	0.040	0.82	0.200	0.150	0.81	5.2
	Testing	0.045	0.80	0.213	0.160	0.79	5.8
	Validation	0.048	0.78	0.219	0.165	0.77	6.0
Bi-GRU-DMN	Training	0.035	0.85	0.187	0.140	0.84	4.7
	Testing	0.038	0.83	0.195	0.148	0.82	5.1

	Validation	0.041	0.81	0.202	0.153	0.80	5.3
RFR	Training	0.042	0.83	0.205	0.155	0.82	5.4
	Testing	0.046	0.80	0.214	0.163	0.79	5.9
	Validation	0.049	0.77	0.221	0.168	0.76	6.2
RNN	Training	0.037	0.84	0.192	0.145	0.83	4.9
	Testing	0.040	0.82	0.200	0.152	0.81	5.4
	Validation	0.043	0.80	0.207	0.156	0.79	5.6
Proposed RBF	Training	0.032	0.87	0.179	0.130	0.86	4.5
	Testing	0.035	0.85	0.187	0.138	0.84	4.8
	Validation	0.038	0.83	0.195	0.144	0.82	5.0

Consistently outperforming current techniques over all datasets, the proposed deep radial basis function (RBF) network shows Table 4 Indicating better accuracy and model fit, the RBF network performs the lowest mean squared error (MSE) of 0.032 and the highest coefficient of determination (R2) of 0.87 for training datasets. With a mean absolute error (MAE) of 0.130 and a root mean squared error (RMSE) of 0.179, the RBF model's mean absolute error (MAE) is lower than those of other methods, therefore showing reduced prediction error. Adjusted R^2 and mean absolute percentage error (MAPE) measurements also help for the RBF network reflection of resilience and accuracy.

For testing and validation datasets, the RBF network maintains leadership with lower MSE, RMSE, and MAE values than ANN, Bi-GRU-DMN, RFR, and RNN. The R^2 values still show the best since the RBF model continuously effectively caputrees the volatility in soil properties. Consequently, the results provide more consistent forecasts than current techniques since the RBF network represents complex nonlinear interactions with great accuracy.

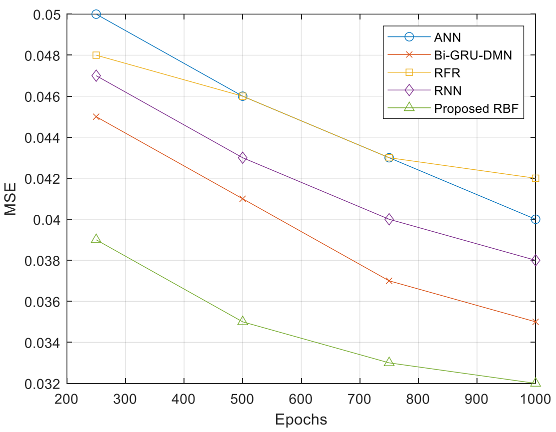

Figure 2: MSE

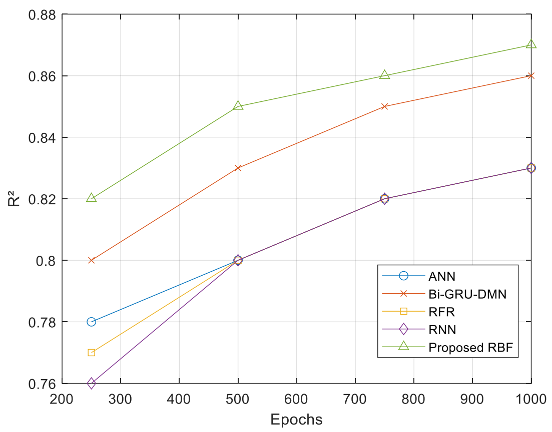

Figure 3: R^2

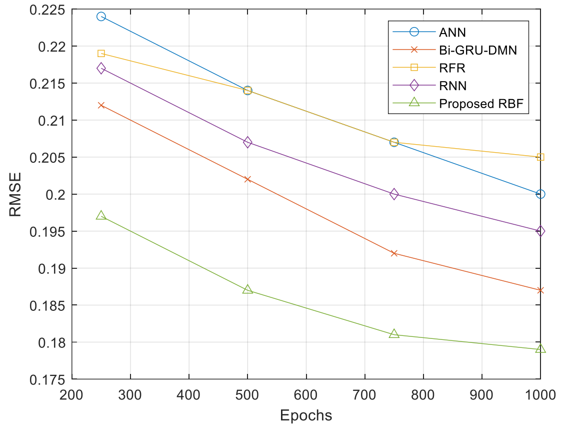

Figure 4: RMSE

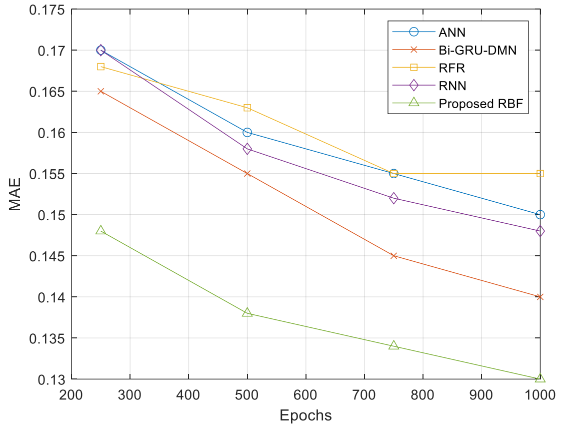

Figure 5: MAE

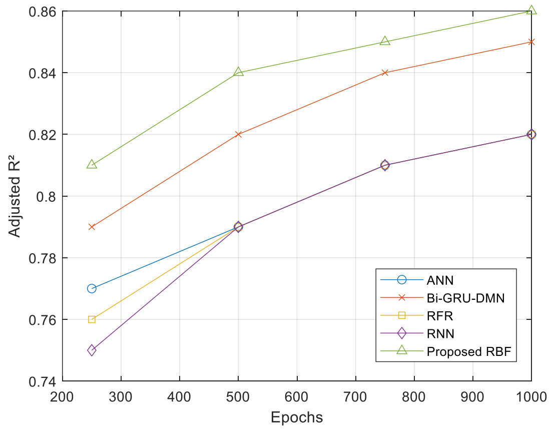

Figure 6: Adjusted R^2

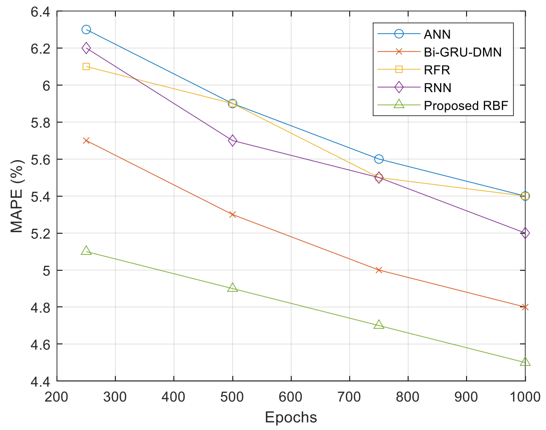

Figure 7: MAPE (%)

The proposed deep radial basis function (RBF) network demonstrates appreciable performance gain in figure 2 – 7 over training epochs, compared to current methods. First at 250 epochs, the RBF network finds mean squared error (MSE) of 0.039, coefficient of determination (R^2) of 0.82, and RMSE of 0.197. Training runs to 1000 epochs helps these values to improve; the RBF network achieves MSE of 0.032, R^2 of 0.87, and RMSE of 0.179. This implies that by gradually caputreing the basic trends in the data, the RBF model reduces prediction errors and increases accuracy. Current methods include ANN, Bi-GRU-DMN, RFR, and RNN show slower convergence and less performance measure improvement with additional epochs. For 1000 epochs, Bi-GRU-DMN for example gets a R^2 of 0.86, still less than the R^2 of the RBF network. Likewise, RNN and ANN models do not show as substantial declines in MSE and RMSE, therefore highlighting the improved capacity of the RBF network to explain complex, nonlinear interactions efficiently with continuous training.

5. Conclusion

In this work, we studied for soil nutrient data classification using nonlinear regression and deep radial basis function (RBF) networks. By means of nonlinear regression techniques, the method captures intricate relationships between soil features acquired from high-resolution remote sensing images, hence improving the quality and relevance of the input characteristics. These features then are sent into a deep RBF network with radial basis functions and many hidden layers to mimic complex, nonlinear patterns. This combined approach far outperforms standard methods including artificial neural networks (ANN), Bi-directional Gated Recurrent Unit with Dynamic Memory Networks (Bi-GRU-DMN), Random Forest Regression (RFR), and Recurrent Neural Networks (RNN), our study found. Therefore, the proposed method offers a robust and precise tool for the classification of soil nutrients, thereby boosting agricultural control and environmental planning.

References

[1] Zhang, Y., Wei, L., Lu, Q., Zhong, Y., Yuan, Z., Wang, Z., ... & Yang, Y. (2023). Mapping soil available copper content in the mine tailings pond with combined simulated annealing deep neural network and UAV hyperspectral images. *Environmental Pollution*, *320*, 120962.

[2] Choudhry, M. D., Sivaraj, J., Munusamy, S., Muthusamy, P. D., & Saravanan, V. (2024). Industry 4.0 in Manufacturing, Communication, Transportation, and Health Care. Topics in Artificial Intelligence Applied to Industry 4.0, 149-165.

[3] Liu, M., Su, W. H., & Wang, X. Q. (2023). Quantitative evaluation of maize emergence using UAV imagery and deep learning. *Remote Sensing*, *15*(8), 1979.

[4] Praghash, K., Yuvaraj, N., Peter, G., Stonier, A. A., & Priya, R. D. (2022, December). Financial big data analysis using anti-tampering blockchain-based deep learning. In *International Conference on Hybrid Intelligent Systems* (pp. 1031-1040). Cham: Springer Nature Switzerland.

[5] Singh, A., Gaurav, K., Sonkar, G. K., & Lee, C. C. (2023). Strategies to measure soil moisture using traditional methods, automated sensors, remote sensing, and machine learning techniques: review, bibliometric analysis, applications, research findings, and future directions. *IEEE Access*, *11*, 13605-13635.

[6] Ramkumar, M., Logeshwaran, J., & Husna, T. (2022). CEA: Certification based encryption algorithm for enhanced data protection in social networks. *Fundamentals of Applied Mathematics and Soft Computing*, *1*, 161-170.

[7] Khatti, J., & Grover, K. S. (2023). Prediction of compaction parameters for fine-grained soil: Critical comparison of the deep learning and standalone models. *Journal of Rock Mechanics and Geotechnical Engineering*, *15*(11), 3010-3038.

[8] Gobinathan, B., Mukunthan, M. A., Surendran, S., Somasundaram, K., Moeed, S. A., Niranjan, P., ... & Sundramurthy, V. P. (2021). A novel method to solve real time security issues in software industry using advanced cryptographic techniques. *Scientific Programming*, *2021*(1), 3611182.

[9] Bawa, A., Samanta, S., Himanshu, S. K., Singh, J., Kim, J., Zhang, T., ... & Ale, S. (2023). A support vector machine and image processing based approach for counting open cotton bolls and estimating lint yield from UAV imagery. *Smart Agricultural Technology*, *3*, 100140.

[10] Singh, A., & Gaurav, K. (2023). Deep learning and data fusion to estimate surface soil moisture from multi-sensor satellite images. Scientific Reports, 13(1), 2251.

[11] Nijaguna, G. S., Manjunath, D. R., Abouhawwash, M., Askar, S. S., Basha, D. K., & Sengupta, J. (2023). Deep learning-based improved WCM technique for soil moisture retrieval with satellite images. Remote Sensing, 15(8), 2005.

[12] Zhang, Y., Han, W., Zhang, H., Niu, X., & Shao, G. (2023). Evaluating soil moisture content under maize coverage using UAV multimodal data by machine learning algorithms. Journal of Hydrology, 617, 129086.

[13] Khosravi, K., Rezaie, F., Cooper, J. R., Kalantari, Z., Abolfathi, S., & Hatamiafkoueieh, J. (2023). Soil water erosion susceptibility assessment using deep learning algorithms. Journal of Hydrology, 618, 129229.

[14] https://www.kaggle.com/datasets/jayaprakashpondy/soil-image-dataset