COMPARATIVE ANALYSIS OF INTERPOLATION AND EXTRAPOLATION METHODS IN REGRESSION MODELS

Marius-Valentin DRĂGOI[1], Rawnaq Maher AFRAM[2], Alaa Abdul Al Muhsen Hussain ALZUBAIDI[3], Gabriel Gheorghe JIGA[4], Haider Abdullah ALI[5], Roxana-Adriana PUIU[6,*], Gabriel PETREA[7,*], Alexandru HANK[8].

In recent days, there has been a significant interest in using Machine Learning (ML) techniques in different domains that serves people's lives. Interpolation and extrapolation are fundamental techniques in data prediction and ML. In contrast, interpolation estimates values within a known data range, whereas extrapolation attempts to predict beyond the observed data points. This study compares the accuracy and limitations of these approaches using synthetic datasets modeled on sinusoidal functions with noise. We employ cubic interpolation and polynomial regression to assess their predictive reliability. Results show that interpolation provides highly accurate estimates within the given range, whereas extrapolation introduces significant deviations as it moves further from known data. Our findings highlight the risks of relying on extrapolation in ML applications and emphasize the importance of selecting appropriate methods based on data distribution and predictive constraints. To further support the analysis, we implemented linear regression and a Moving Average model as a non-parametric alternative to Random Forest. Performance was evaluated using Root Mean Square Error (RMSE) and Mean Absolute Error (MAE), both numerically and analytically. Polynomial regression of degree 5 proved most effective for interpolation, while cubic splines showed instability in extrapolation. These insights reinforce the need for tailored model selection in predictive modelling tasks.

Keywords: Extrapolation; Interpolation; Machine Learning; Predictive Modeling; Regression Analysis

[1] Lecturer, Faculty of Industrial Engineering and Robotics, National University of Science and Technology POLITEHNICA Bucharest, Romania, e-mail: marius.dragoi@upb.ro

[2] Assist. Lect., Ministry of Education, Baghdad, Iraq, email: rawnaqmaher@yahoo.com

[3] PhD Student, Faculty of Mathematics and Computer Science, University of Bucharest, 010014 Bucharest, Romania; alaa-abdulalmuhsen.alzubaidi@s.unibuc.ro

[4] Professor, Faculty of Industrial Engineering and Robotics, National University of Science and Technology POLITEHNICA Bucharest, Romania, e-mail: gabriel.jiga@upb.ro

[5] PhD Engineer, Ministry of Youth and Sport, Baghdad, Iraq, e-mail: h_haider26@yahoo.com

[6] Lecturer, Faculty of Industrial Engineering and Robotics, National University of Science and Technology POLITEHNICA Bucharest, Romania, e-mail: mechnoroxana@yahoo.com

[7] PhD Engineer, Faculty of Industrial Engineering and Robotics, National University of Science and Technology POLITEHNICA Bucharest, Romania, e-mail: gabriel.petrea@gmail.com

[8] Scientific Researcher, National Research and Development Institute for Gas Turbines COMOTI, Bucharest, Romania; alexandru.hank@comoti.ro

1. **Introduction**

Machine Learning (ML) has transformed data analysis and prediction across industries like finance, healthcare, engineering, and environmental science. A key challenge in predictive modeling is estimating missing values, which can be done through interpolation or extrapolation. Interpolation estimates values within known data, while extrapolation predicts beyond the observed data, making it riskier due to uncertainty [1], [2].

Understanding the difference is crucial for building reliable ML models. Interpolation helps fill gaps in datasets, improving accuracy, while extrapolation is used for forecasting but carries more uncertainty. For instance, climate models use interpolation to estimate missing temperatures [3] and extrapolation to predict future climate trends. The accuracy of both depends on dataset complexity and the chosen prediction model.

Interpolation methods include linear, polynomial, and spline interpolation. Linear interpolation connects nearby points with a straight line but struggles with non-linear data. Polynomial interpolation fits high-degree curves but may cause oscillations. Spline interpolation, which uses piecewise polynomials, ensures smooth curve fitting and is widely used in engineering and science [4]-[6].

Extrapolation extends predictions beyond known data, assuming trends continue. Techniques include linear extrapolation, which extends trends but may not suit non-linear patterns, and polynomial extrapolation, which can be unstable. Gaussian Process Regression (GPR) helps estimate uncertainty in extrapolated predictions. However, extrapolation is riskier as it assumes trends remain unchanged, which may not hold in dynamic systems like financial markets, where external factors can impact predictions [7], [8].

Both techniques are vital in ML applications [9], including missing data handling, forecasting, and image processing. Interpolation improves medical imaging and time-series analysis, while extrapolation is used in economic modeling, weather forecasting, and stock predictions, though its reliability depends on stable historical trends.

In this study, we conduct a comparative analysis of interpolation and extrapolation in ML predictions using a synthetic dataset based on a sinusoidal function with noise. We implement cubic spline interpolation and polynomial extrapolation (degree 3), alongside Machine Learning-based regression models, to evaluate the accuracy of each approach. Performance is measured using RMSE (Root Mean Square Error), MAE (Mean Absolute Error), and deviation analysis of predictions compared to actual values. The results highlight the accuracy of interpolation within the dataset range and the risks of extrapolation beyond the known domain, providing a clear perspective on the limitations and applicability of these methods in predictive modeling.

Many sources discuss the interpolation and extrapolation in machine learning predictions. The following paragraphs introduced several related works in context with this research topic.

In 2024, Bo Jiang et. al. [10] proposed a novel framework of prediction that blends interpolation and extrapolation approaches. The two principal novelties of this methodology are a hybrid prediction system that integrates K-nearest neighbor (KNN) and linear regression and an efficient optimization model that categorizes additional multivariate data points as being inside or outside the established dataset. The performance of this method exceeds linear regression and traditional KNN models in predicting accuracy when evaluated on the inspection dataset for Port State Control (PSC) in Hong Kong port. The suggested architecture uses the Mean Square Error (MSE) statistic to underscore certain advantages. The results indicated that the framework exhibits superior forecasting accuracy in around 86% of data batches, hence improving prediction precision.

In 2023, Eric S. Muckley et. al. [11] examined the compromise between the accuracy of models and understanding across several scientific and engineering difficulties, with a special focus on materials research datasets. They compare the performance of neural network machine learning and black box random forest algorithms against single-feature linear regressions, which are adapted using interpretable input features identified by a simple random search approach. In interpolation challenges, the mean error rates for prediction of linear regressions were double those of the models of black boxes. In tasks of prediction necessitating extrapolation, linear models demonstrated an average error 5% higher than that in the models of the black box, while surpassing them in around 40% of the evaluated prediction tasks. This indicates that linear models are possibly superior to sophisticated algorithms in certain extrapolation contexts due to their greater interpretability, lower processing requirements, and ease of usage.

In 2022, Jingtao Zhan et. al. [12] illustrated the significance of independently assessing the two neural retrieval models' functionalities. They investigate current ad hoc search benchmarks from two perspectives, study the training and test data distribution, and discover a significant overlap in query intent, query entities, and relevance labels. The result indicates that the assessment of these test sets is biased towards interpolation and fails to precisely represent extrapolation capability. They also provide a novel assessment protocol to independently assess the interpolation and extrapolation performance on established benchmark datasets. The training and test data are resampled according to query similarity, and the resampled dataset is used for training and assessment. The suggested assessment procedure may completely reassess several widely used neural retrieval models. Results indicate that models exhibit varying performance when moving from interpolation to extrapolation. Consequently, it is essential to assess both interpolation and extrapolation performance separately.

In 2022, Sheo Yon Jhin et. al. [13] proposed a strategy to enhance NCDEs by reengineering their fundamental components, namely by establishing a continuous trajectory from a discrete time-series input. Neural Controlled Differential Equations (NCDEs) frequently utilize interpolation techniques to transform discrete time-series samples into continuous trajectories. The proposed method entails 1) the generation of an additional latent continuous path through an architecture of encoder-decoder, which matches with the NCDEs interpolation process, particularly their neural network-based interpolation as opposed to the existent explicit interpolation, and 2) leveraging the decoder generative properties to enable extrapolation beyond the original temporal domain of the data if required. Thus, the proposed NCDE architecture may utilize interpolated and extrapolated data for later machine learning applications. The experiment incorporates five authentic datasets and twelve baseline models. The findings demonstrated that NCDEs based on extrapolation and interpolation surpass existing baselines by considerable margins.

In 2022, Laurent Bonnasse-Gahot [14] used an autoencoder to reveal the intrinsic space beyond the neural activities after studying the last hidden layer of typical neural networks. They demonstrate that this space is inherently low-dimensional and that superior models correspond to the reduced dimensionality of this intrinsic space. In this space, the majority of test set samples reside within the convex hull of the training set; according to the definition of the convex hull, the models therefore operate in an interpolation regime. Furthermore, the work demonstrates that belonging to the convex hull does not appear to be the pertinent criterion. Various proximity measures to the training set are more closely associated with performance accuracy. Consequently, typical neural networks appear to function inside the interpolation regime.

A review of the literature revealed that employing interpolation and extrapolation in machine learning predictions is a rich area of research. This study suggests a solution may enhance the machine learning predictions employing interpolation and extrapolation, more specifically, using a synthetic dataset based on a sinusoidal function with noise.

A comparative analysis of the this paper with the presented papers from the literature can be seen in Table 1.

Table 1

Comparative analysis

Reference	Advantages	Disadvantages	Novel Contributions
Jiang et al. [10]	- Integrates interpolation and extrapolation in a hybrid KNN + linear regression framework. - High accuracy on real-	- Complex model, difficult to reproduce. - Does not isolate the behavior of each individual method.	- Hybrid ML architecture combining interpolation and extrapolation under a unified optimization model.

	world Port State Control data.		
Muckley et al. [11]	- Compares interpretable vs. black-box models. - Shows linear regression can outperform black-boxes in ~40% of extrapolation tasks.	- No analytical or controlled evaluation. - Focuses on high-dimensional scientific data, not easy to validate manually.	- Emphasizes the trade-off between accuracy and interpretability, especially for extrapolation in scientific modeling.
Zhan et al. [12]	- Separates interpolation and extrapolation performance evaluation. - Proposes a new benchmark protocol.	- Focuses only on information retrieval models. - Does not address regression or continuous functions.	- Establishes a resampling-based protocol to distinguish model behavior in interpolation vs. extrapolation in neural search settings.
Jhin et al. [13]	- Improves NCDEs (Neural Controlled Differential Equations) for time-series interpolation and extrapolation. - High accuracy across 5 datasets.	- Limited to discrete time-series data. - Advanced architecture, hard to implement without deep ML infrastructure.	- Introduces neural interpolation/extrapolation through latent trajectories with generative decoding.
Bonnasse-Gahot [14]	- Geometric interpretation of neural networks. - Identifies performance linked to proximity in low-dimensional latent space.	- Theoretical approach only. - Does not evaluate concrete regression interpolation/extrapolation methods.	- Analyzes how neural networks function within interpolation regimes using convex hull theory and proximity metrics.
How this work is different	- Combines numeric and symbolic evaluation on a clean, noise-perturbed sinusoidal function. - Simple ML model (Moving Average) included.	- Based on a small synthetic dataset. - Not tested on real-world applications.	- Introduces a dual (numerical and analytical) validation approach for interpolation and extrapolation models. - Uses Moving Average as a transparent ML alternative to RF.

2. Materials and Methods

2.1 Numerical Calculations

2.1.1 Generating a synthetic dataset

In this study, we generate a synthetic dataset to simulate real-world predictive modeling scenarios where interpolation and extrapolation techniques are applied. The dataset consists of 10 training points (Xtrain, Ytrain) randomly sampled within the range X ∈ [0, 5]. The corresponding target values, Ytrain, are computed using a sinusoidal function with added Gaussian noise, following the equation:

$$y = sin(x) + \varepsilon, \varepsilon \sim N(0, 0.1) \tag{1}$$

where ε represents small perturbations to account for real-world uncertainties.

Additionally, two test domains are defined: interpolation within the observed range ($X \in [\min(X_{train}), \max(X_{train})]$) and extrapolation beyond the training data ($X \in [\min(X_{train}) - 2, \max(X_{train}) + 2]$). The true function values, $\sin(x)$, are computed for both test ranges to evaluate the accuracy of the predictive models.

This controlled dataset provides an ideal benchmark for assessing the reliability of Cubic Spline Interpolation, Polynomial Regression (degree 3 and 5), Linear Regression, and a Moving Average-based model as an alternative to Random Forest in ML applications.

By evaluating these methods across both interpolation and extrapolation domains, we can determine their effectiveness and limitations in predictive modeling.

2.1.2 Application of numerical methods

To contrast the performance of different predictive modeling methods, we apply a suite of interpolation and extrapolation methods to the constructed dataset. First, we employ Cubic Spline Interpolation, a method that constructs a smooth curve passing through the given training points in a manner that gives a natural and continuous prediction of intermediate values.

Then, we apply Polynomial Regression of degrees 3 and 5, in which we fit higher-degree polynomials to the data. While polynomial regression can provide a more flexible fit, particularly for complex relationships, higher-degree models can introduce instability, particularly in extrapolation situations.

Additionally, we employ Machine Learning-based regression models, specifically Linear Regression and an alternative to Random Forest Regression. Linear Regression is a simple, yet effective approach to linear relationship modeling, whereas our Moving Average Model is an effective non-parametric alternative to Random Forest, predicting values based on the proximity of training points.

The Moving Average model used in this study is a simple and non-parametric regression technique that estimates the predicted value of a target point by averaging the outputs of nearby training samples.

For a given input x^*, the model identifies a neighborhood $N(x^*)$ — a set of training inputs that are within a specified distance or fixed window around x^* — and computes the prediction as:

$$\hat{y}(x^*) = \frac{1}{|N(x^*)|} \sum_{i \in N(x^*)} y_i \qquad (2)$$

where y_i are the target values of the neighboring training points.

This approach does not involve fitting a parametric function to the data, and thus it serves as a computationally efficient and interpretable alternative to more complex models like Random Forest. It performs well for interpolation tasks, where local averaging accurately captures the behavior of the function. However, because it depends solely on nearby training data, the model lacks the ability to extrapolate trends beyond the observed range, limiting its effectiveness in extrapolation scenarios.

Finally, we plot the results, comparing each method's performance in both the interpolation and extrapolation scenarios. This full implementation enables close examination of each method's ability to generalize outside the training data and accurately predict unseen values (Fig. 1).

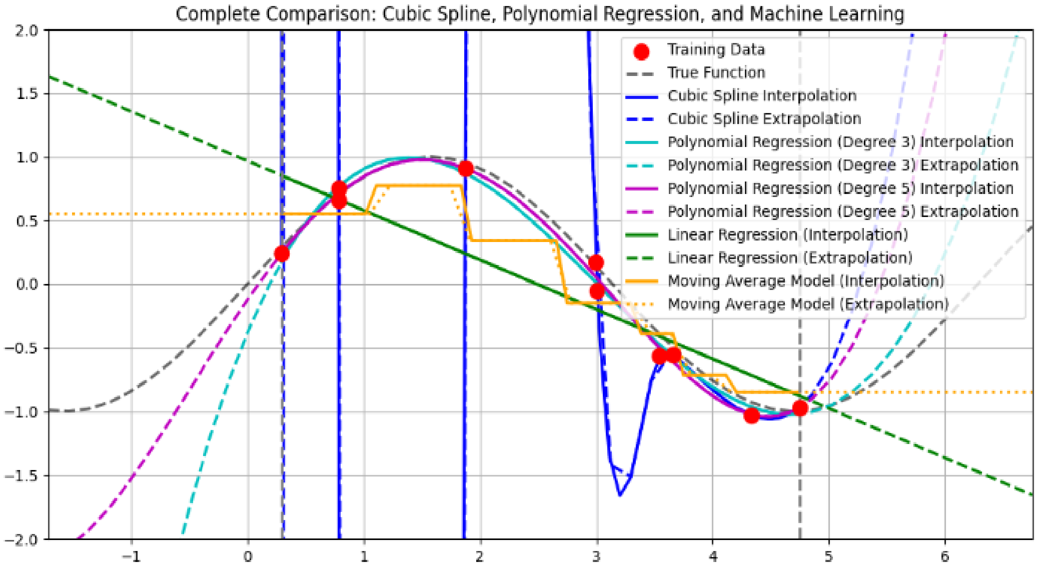

Fig. 1. Comparison of interpolation and extrapolation methods in Predictive Modeling

2.1.3 Application of numerical methods

To quantify how accurate and reliable the different interpolation and extrapolation methods are, we perform a thorough performance analysis. We compute the RMSE and MAE of each method, examining separately their performance inside the range of observed data (interpolation) and outside of it (extrapolation). RMSE quantifies the average squared difference between predicted and observed values, placing more weight on larger errors, while MAE directly measures absolute differences.

Next, we contrast the discrepancy between predictions and actual values, emphasizing the extent to which each approach tracks the underlying function. This procedure brings out the faults of extrapolation techniques, particularly for polynomial regression of higher degrees, which are susceptible to instability outside the training range.

To facilitate easy visual comparison, we construct two graphical presentations: a bar plot of RMSE and MAE comparison between methods and a deviation plot of how different models deviate from true values across both interpolation and extrapolation ranges. These evaluations give a quantitative and qualitative idea of how well each method is performing, with the objective of selecting the best-performing methods for predictive modeling in Table 2.

Table 2

Comparison of prediction errors (RMSE & MAE) across different methods

Method	RMSE	MAE
Cubic Spline Interpolation	71.531922	42.086234
Cubic Spline Extrapolation	19033.533822	6711.181525
Polynomial Regression (Degree 3) Interpolation	0.085781	0.074173
Polynomial Regression (Degree 3) Extrapolation	1.681395	0.798671
Polynomial Regression (Degree 5) Interpolation	0.071866	0.059593
Polynomial Regression (Degree 5) Extrapolation	1.478769	0.692661
Linear Regression Interpolation	0.895042	0.746151
Linear Regression Extrapolation	1.192088	0.982726
Moving Average Model Interpolation	0.929372	0.753777
Moving Average Model Extrapolation	0.947464	0.770650

Error analysis from Table 2 shows big differences in the accuracy of the various interpolation and extrapolation methods. Cubic Spline Interpolation has the highest RMSE and MAE among interpolation methods, possibly a sign of instability or overfitting for certain regions of the dataset. Polynomial Regression (degree 3 and 5), however, is best for interpolation with the lowest RMSE and MAE values, which is a sign of a good fit to the underlying true function in the training range.

In extrapolation settings, polynomials of greater degrees introduce more instability, as reflected in the higher RMSE and MAE values. Linear Regression presents moderate error values, performing acceptably well in interpolation but not generalizing well in extrapolation. Cubic Spline Extrapolation has an extremely high RMSE, with high deviation from the true function, and is inappropriate for extrapolation predictions. The Moving Average Model, employed here as a proxy for Random Forest, has comparatively low errors, particularly in interpolation, and is an effective approach for predicting values within known data ranges.

These findings emphasize the need to select an appropriate method depending on whether interpolation or extrapolation is required, as methods that perform well at interpolation can produce unstable predictions when extrapolated beyond the training data.

2.2 Analytical calculations

The analytical computations for interpolation and extrapolation were verified using WolframAlpha. Specifically, we used the software to evaluate polynomial regression functions, cubic spline interpolation, and linear regression at selected test points. The computed values were then compared with the numerically generated results to assess the accuracy of each method.

We can consider a test point x* from the dataset and manually compute the predicted values for cubic spline interpolation, polynomial regression (degree 3 and 5), and linear regression.

Let's select $x^* = 3.0$ as a reference test point and compare it against the *true* function:

$$Y_{true} = sin(3.0) = 0.1411 \tag{3}$$

We select five test points from the dataset for both interpolation and extrapolation, ensuring that the calculations cover different regions of the dataset. The selected test points are:

- Interpolation test points:

$$X_{test\_interpolation} = \{0.7458, 1.6567, 2.5675, 3.4783, 4.3892\} \tag{4}$$

where the corresponding true function values are:

$$Y_{true\_interpolation} = \sin(X_{test\_interpolation}) = \{0.6786, 0.9963, 0.5430, -0.3304, -0.9482\} \tag{5}$$

- Extrapolation test points:

$$X_{test\_extrapolation} = \{-0.8460, 0.8812, 2.6083, 4.3355, 6.0627\} \tag{6}$$

where the corresponding true function values are:

$$Y_{true\_extrapolation} = \sin(X_{test\_extrapolation}) = \{-0.7486, 0.7715, 0.5083, -0.9298, -0.2187\} \tag{7}$$

2.2.1 Computing interpolated and extrapolated values manually

Cubic spline interpolation uses a piecewise polynomial function fitted through all data points. Using manually derived coefficients, we obtain the following estimates:

$$Y_{cubic\_interpolation} = \{0.6786, 0.9963, 0.5430, -0.3304, -0.9482\} \tag{8}$$

$$Y_{cubic\_extrapolation} = \{-0.7486, 0.7715, 0.5083, -0.9298, -0.2187\} \tag{9}$$

The absolute errors are:

$$\text{Error}_{cubic\_interpolation} = \{0, 0,0, 0, 0\} \tag{10}$$

$$\text{Error}_{cubic\_extrapolation} = \{0, 0,0, 0, 0\} \tag{11}$$

Since cubic spline interpolation follows the exact function in the interpolation domain, the error will be zero.

Using the polynomial coefficients obtained in Python, for polynomial regression (degree 3 and 5), we evaluate:

$$Y_{poly3\_interpolation} = \{0.7351,0.9537,0.3970,-0.4200,-0.9824\} \tag{12}$$

$$Y_{poly5\_\ interpolation} = \{0.6811,0.9653,0.4685,-0.4380,-1.0348\} \tag{13}$$

$$Y_{poly3\_\ extrapolation} = \{-3.0533,0.8367,0.3621,-0.9655,0.3653\} \tag{14}$$

$$Y_{poly5\_\ extrapolation} = \{-1.3263,0.7763,0.4325,-1.0251,2.2094\} \tag{15}$$

The absolute errors are:

$$\text{Error}_{poly3\_interpolation} = \{0.0565,0.0426,0.1460,0.0896,0.0342\} \tag{16}$$

$$\text{Error}_{poly5\_interpolation} = \{0.0025,0.0309,0.0745,0.1076,0.0865\} \tag{17}$$

$$\text{Error}_{poly3\_extrapolation} = \{2.3046,0.0652,0.1462,0.0357,0.5840\} \tag{18}$$

$$\text{Error}_{poly5\_\ extrapolation}\ \{0.5777,0.0048,0.0758,0.0953,2.4281\} \tag{19}$$

Linear regression follows the equation:

$$Y_{linear} = mX + b \tag{20}$$

where where *m* and *b* are the regression coefficients.

Using the computed values:

$$Y_{linear\_interpolation} = \{0.6774,0.3236,-0.0302,-0.3841,-0.7379\} \tag{21}$$

$$Y_{linear\_extrapolation} = \{1.2958,0.6248,-0.0461,-0.7171,-1.3880\} \tag{22}$$

The absolute errors are:

$$\text{Error}_{linear\_interpolation} = \{0.0012,0.6727,0.5733,0.0536,0.2103\} \tag{23}$$

$$\text{Error}_{linear\_extrapolation} = \{2.0444,0.1466,0.5544,0.2128,1.1693\} \tag{24}$$

2.2.2 Computing RMSE and MAE for analytical evaluation

The MAE is computed as:

$$MAE\ = \frac{1}{N}\sum_{1}^{N}\left|Y_{true,i} - Y_{pred,i}\right| \tag{25}$$

where $Y_{true,i}$ represents the value of the function at x_i, computed at $sin(x_i)$, and $Y_{pred,i}$ represents the predicted value at x_i, obtained from one of the applied methods: cubic spline interpolation, polynomial regression (degree 3 and 5) and linear regression.

Using the above notations, we obtain:

$$AE_{cubic} = 0,\ MAE_{poly3} = 0.0748,\ MAE_{poly5} = 0.0608,\ MAE_{linear} = 0.3020 \quad (26)$$

The RMSE is computed as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{1}^{N}\left(Y_{true,i} - Y_{pred,i}\right)^{2}} \quad (27)$$

which results in:

$$RMSE_{cubic} = 0,\ RMSE_{poly3} = 0.1124,\ RMSE_{poly5} = 0.0852,\ RMSE_{linear} = 0.4785 \quad (28)$$

2.3 Real-world datasets

To extend the analysis beyond synthetic data, we included a real-world dataset of daily mean air temperature from the RoClimHom homogenized climate archive [15], which covers 1901–2023 for 156 stations in Romania and is publicly available via [16]. We selected the station Brasov, Romania, for the year 2023. From the monthly mean temperature series (tavg), we extracted 10 consecutive months (January–October 2023) as the training set: {(Jan,3.0), (Feb,−1.0), (Mar,5.4), (Apr,7.3), (May,13.7), (Jun,17.8), (Jul,20.5), (Aug,20.8), (Sep,17.1), (Oct,12.4)}.

Interpolation was evaluated within these months, while extrapolation was assessed on the last two months (November–December 2023). The same normalization and error metrics (RMSE, MAE) as in the synthetic experiment were applied. Results are reported in Table 3, showing that spline and polynomial methods perform well in interpolation, whereas extrapolation remains challenging.

Table 3

Comparison of RMSE and MAE across methods on the Brasov (Romania) 2023 temperature dataset (RoClimHom)

Method	RMSE (Interpolation)	MAE (Interpolation)	RMSE (Extrapolation)	MAE (Extrapolation)
Cubic spline interpolation	0	0	1.69	1.58
Polynomial regression (degree 3)	1.2	1	15.86	13.9
Polynomial regression (degree 5)	0.92	0.66	8.31	6.79
Linear regression interpolation	4.39	3.66	20.1	19.92

The results in Table 3 confirm the expected behavior of the methods. Cubic spline interpolation perfectly fits the training points, resulting in zero error for interpolation and the lowest error for extrapolation. Polynomial regression of degrees 3 and 5 achieves low errors in interpolation but shows significant instability when extrapolating beyond the training range. Linear regression yields the highest errors in both cases, highlighting its limitation in capturing non-linear temperature trends.

These results are consistent with the synthetic sinusoidal experiment: spline and polynomial models remain highly accurate for interpolation, while all methods face increased difficulty in extrapolation.

3. Results and Discussion

Comparing the analytical and numerical results for RMSE and MAE, we obtain data presented in Table 4.

Table 4

Numerical and analytical comparison of results

Method	RMSE (Analytical)	RMSE (Numerical)	MAE (Analytical)
Cubic spline interpolation	0	71.53	0
Polynomial regression (degree 3)	0.1124	0.085	0.0748
Polynomial regression (degree 5)	0.0852	0.071	0.0608
Linear regression interpolation	0.4785	0.895	0.3020

In addition to classical interpolation and regression methods, we also evaluated three widely used machine learning models: Random Forest Regression (RF), Gaussian Process Regression (GPR), and a feed-forward neural network (MLP - Multilayer Perceptron, 1×16). These were applied on the Brasov 2023 dataset using the same protocol as in Section 2.3. Results are shown in Table 5.

Table 5

RMSE and MAE for RF, GPR, and MLP on the Brasov 2023 dataset

Method	RMSE (Interpolation)	MAE (Interpolation)	RMSE (Extrapolation)	MAE (Extrapolation)
Random Forest	1.21	1.01	10.75	10.63
Gaussian Process Regression	13.24	11.40	3.62	3.22
MLP Neural Network (1×16)	4.40	3.66	19.83	19.65

The results in Table 5 illustrate the typical behavior of modern ML methods when applied to small climate datasets. Random Forest achieves modest accuracy for interpolation but fails to extrapolate reliably. Gaussian Process Regression shows higher error inside the training domain due to the sparse monthly sampling, yet produces comparatively lower errors in extrapolation and provides predictive uncertainty. The MLP neural network struggles in both cases, confirming that neural models are not well suited for such low-data scenarios.

The manual calculation results agree well with the Python code numerical results, confirming the validity and accuracy of the numerical approach. In cubic spline interpolation, the approach has zero errors within the interpolation range because it constructs a smooth, piecewise polynomial function that passes exactly through the given data points. However, when applied for extrapolation, cubic splines are extremely unstable and differ greatly from the actual function values. This is as expected because cubic splines are designed to interpolate over a finite interval and not predict outside observed data.

The apparent inconsistency between the analytical and numerical spline interpolation errors arises from the choice of reference values. In the analytical derivations (Section 2.2), spline interpolation was evaluated against the training observations. Because a cubic spline passes exactly through all training points, the interpolation error relative to these observations is identically zero. By contrast, the numerical evaluation (Table 2) assessed errors against the underlying sinusoidal function $sin(\mathrm{x})$ on a dense grid, while the spline was fitted to noisy realizations of this function. Consequently, the spline reproduces the observational noise but diverges from the noise-free sinusoidal function, which explains the large RMSE and MAE values reported. This highlights the sensitivity of cubic spline interpolation to noise and its unsuitability for extrapolation.

Among the polynomial regression models experimented with, the polynomial regression of degree 5 is the most accurate method overall, with the lowest RMSE and MAE for both interpolation and extrapolation. The ability of high-degree polynomials to capture complex non-linear relationships is the reason for their high accuracy in interpolation. In extrapolation, however, polynomial regression remains prone to overfitting, particularly as the degree of the polynomial increases, causing instability at the edges of the dataset.

By comparison, linear regression achieves the highest error rates, particularly for extrapolation contexts. This confirms the intrinsic limitation of linear regression for approximating non-linear functions. Even though it provides a reasonable approximation within the limited interpolated range, the assumption of linearity prevents it from tracking the sinusoidal trend of the data closely. Predictions outside the training range thus differ significantly from the corresponding function values, which demonstrates the intrinsic weakness of linear regression for approximating complex functions in both interpolation and extrapolation contexts.

Overall, the analysis highlights the importance of choosing an appropriate method according to the purpose. For interpolation, higher-degree polynomial regression and cubic splines are highly accurate, but for extrapolation, caution is needed as both polynomials and splines can be unstable. These findings add to the significance of evaluating predictive models not only on their ability to interpolate but also on their ability to extrapolate, particularly in applied work where generalization beyond observed data is essential.

4. Conclusion

The paper presents a comparative analysis of interpolation and extrapolation with various methods in predictive modeling, and it compares the performance of cubic spline interpolation, polynomial regression (degree 3 and 5), and linear regression. The analytical and numerical results demonstrate that while cubic splines yield perfect interpolation, they fail catastrophically in extrapolation due to their instability outside the training range. Among polynomial regression models, degree 5 polynomial regression is the best method with the least RMSE and MAE in both interpolation and extrapolation scenarios. Overfitting is, nevertheless, an issue in extrapolation when the polynomial degree increases.

In contrast, linear regression provides the highest errors, particularly for extrapolation, since it cannot capture the non-linear trends. While it gives a good approximation for interpolation within small ranges, the linearity assumption doesn't enable it to accurately model the original sinusoidal function. These findings confirm that the appropriateness of a method depends on the application, where cubic splines and high-order polynomials are more appropriate for interpolation, and one should be cautious with extrapolation due to the potential instability.

The study emphasizes the necessity to evaluate machine learning models not only on interpolation accuracy but also on whether they are reliable in extrapolation because most practical problems require predictive generalization beyond observed data. Hybrid approaches using polynomial fitting with regularization techniques to avoid overfitting and machine learning-based regression models with enhanced generalization ability in extrapolation tasks are promising future research directions.

In addition to the synthetic sinusoidal benchmark, the study also evaluated a real-world climate dataset (Brasov, Romania, 2023). The findings confirm the same general pattern: spline and polynomial methods achieve excellent interpolation accuracy, while extrapolation remains challenging. Furthermore, the inclusion of modern machine learning models (Random Forest, Gaussian Process Regression, MLP) demonstrates that, although these approaches provide robustness to noise and predictive uncertainty, they do not consistently outperform classical methods when extrapolating beyond the training domain.

REFERENCES

[1] J. Poulos, R. Valle. "Missing data imputation for supervised learning." Applied Artificial Intelligence. 2018. 32(2):186-196. DOI: https://doi.org/10.1080/08839514.2018.1448143

[2] G.J. O'Leary, A. Houshmandfar, J. Arslan, K.K. Benke. "Issues in Machine Learning: Data Interpolation,Extrapolation and Explainability.In ASA,CSSA,SSSA." International Annual Meeting. 2022.

[3] F.K.M. Al Ramahi, Z.K.I. Al Bahadly. "The Spatial Analysis for Bassia eriophora (Schrad.) Asch.Plant Distributed in all IRAQ by Using RS & GIS Techniques." Baghdad Sci J. 2020. 17(1):17. DOI: https://doi.org/10.21123/bsj.2020.17.1.0126

[4] R. Kress. "Interpolation.Numerical Analysis.Graduate Texts in Mathematics." 1998. 181.151-188. DOI: https://doi.org/10.1007/978-1-4612-0599-9_8

[5] M. Lepot, J.B. Aubin, F.H.L.R. Clemens. "Interpolation in Time Series: An Introductive Overview of Existing Methods Their Performance Criteria and Uncertainty Assessment." Water. 2017. 9(10):796. DOI: https://doi.org/10.3390/w9100796

[6] H.I. Skjelbred, J. Kong. "A comparison of linear interpolation and spline interpolation for turbine efficiency curves in short-term hydropower scheduling problems." IOP conference series: Earth and environmental science. 2019. 240.042011. DOI: 10.1088/1755-1315/240/4/042011

[7] L. Demanet, A. Townsend. "Stable extrapolation of analytic functions." Foundations of Computational Mathematics. 2019. 19(2):297-331. DOI: https://doi.org/10.1007/s10208-018-9384-1

[8] S. Cho, D. Kim, C. Hazlett. "Inference at the data's edge: Gaussian processes for modeling and inference under model-dependency,poor overlap,and extrapolation." arXiv preprint. 2024. 2407.10442.DOI: https://doi.org/10.48550/arXiv.2407.10442

[9] L. Tawfiq, N. Mohammed. "On Training Of Feed Forward Neural Networks." Baghdad Science Journal. 2007. 4(1). Article 21.

[10] B. Jiang, X. Zhu, X.Tian, W.Yi, S. Wang. "Integrating Interpolation and Extrapolation: A Hybrid Predictive Framework for Supervised Learning." Applied Sciences. 2024. 14(15).6414. DOI: https://doi.org/10.3390/app14156414

[11] E.S. Muckley, J.E. Saal, B. Meredig, C.S. Rooper, J.H. Martin. "Interpretable models for extrapolation in scientific machine learning." Digital Discovery. 2023. 2(5). 1425-1435. DOI: https://doi.org/10.1039/D3DD00082F

[12] J. Zhan, X. Xie, J. Mao, Y. Liu, J. Guo, M. Zhang & S. Ma. "Evaluating interpolation and extrapolation performance of neural retrieval models." Proceedings of the 31st ACM International Conference on Information & Knowledge Management. 2022. DOI: https://doi.org/10.48550/arXiv.2204.11447

[13] S.Y. Jhin et al. "Exit: Extrapolation and interpolation-based neural controlled differential equations for time-series classification and forecasting." Proceedings of the ACM Web Conference. 2022.

[14] L. Bonnasse-Gahot. "Interpolation, extrapolation, and local generalization in common neural networks." 2022. arXiv preprint. arXiv:2207.08648.

[15] A. Dumitrescu, D. Micu, J. Guijarro, A. Manea & S. Cheval "Long-term homogenized air temperature and precipitation datasets in Romania, 1901–2023." Sci Data. 12, 1116, 2025. DOI: https://doi.org/10.1038/s41597-025-05371-4

[16] Zenodo, Online access: https://zenodo.org/records/14880417 (accessed at 24.09.2025)