Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
DOI : 10.47974/JDMSC-2518

ML techniques in polynomial algebra for advanced signal processing

Chithirai Pon Selvan §
School of Science and Engineering
Curtin University
345031
Dubai

Swaroop Ramaswamy †
Department of Electrical Engineering
Amity University
Dubai
U.A.E.

Aakash Shivaji Yadav *
Department of Management
Institute of Management and Entrepreneurship Development
Bharati Vidyapeeth (Deemed to be University)
Pune 411038
Maharashtra
India

Ulugbek Bobamuratov ‡
Department of Information Technology and Exact Sciences
Termez University of Economics and Service
Termez 190111
Uzbekistan

§ *E-mail:* pon.selvan@curtindubai.ac.ae
† *E-mail:* spillai@amityuniversity.a
* *E-mail:* aakash.yadav@bharatividyapeeth.edu (Corresponding Author)
‡ *E-mail:* ulugbek_bobamuratov@tues.uz

Anjali V. Pise [@]
Department of Computer Science & Engineering
Tulsiramji Gaikwad Patil College of Engineering and Technology
Nagpur 441108
Maharashtra
India

Alaknanda S. Patil [#]
Department of Electronics and Telecommunication
Bhivarabai Sawant College of Engineering and Research, Polytechnic
Pune 411041
Maharashtra
India

Abstract

This study looks at how machine learning (ML) and polynomial math can be used together to improve advanced signal processing. The study creates new mathematical models that use the polynomial structures that are already present in signal data. These models combine vector metric spaces and orthogonal polynomial sequences to find features and show signals. The suggested way is based on quickly getting polynomial features that keep the purity of the signal and allow for strong learning. Combining machine learning algorithms with polynomial algebraic structures makes it easier to analyse and handle complex signals. This solves problems with convergence and orthogonality in very large vector spaces. The method offers better accuracy in jobs like signal classification, noise reduction, and rebuilding.

Subject Classification: *31A10, 33E30.*

Keywords: *Polynomial algebra, Signal processing, Machine learning, Orthogonal sequences, Feature extraction.*

I. Introduction

Signal processing is very important in many areas, such as biological engineering, computer systems, telecommunications, and picture and sound analysis. Most of the time, linear models and classical changes are used in traditional signal processing methods. These can be limiting when dealing with complicated, nonlinear, or high-dimensional signal data. With its deep mathematical structure, polynomial algebra gives us strong tools for describing and analysing these kinds of signals, especially when we are working with nonlinearity and complicated signal behaviours [1]. Polynomial algebra plays a vital role in computational mathematics, offering a systematic framework for

[@] *E-mail:* **avm.pise@gmail.com**
[#] *E-mail:* **patilalaknanda@yahoo.com**

representing and manipulating polynomial expressions. It supports problem-solving in diverse fields such as coding theory, cryptography, and numerical analysis by enabling efficient operations on large datasets. Its structured approach aids in modeling, optimization, and the development of algorithms for real-world applications. New developments in machine learning (ML) have changed the way data is analysed, making signal processing more flexible, accurate, and automatic. However, combining ML and polynomial math to use their united skills is still an area that hasn't been looked into very much [2, 3]. ML models can better represent features when they use orthogonal polynomial sets and vector measure spaces, which are natural ways to store signal traits. These mathematics structures make it easier for efficient convergence qualities and orthogonality to work, which are necessary to cut down on duplicate information and make learning more stable [4]. The goal of this study is to create a more advanced structure for signal processing by bridging the gap between polynomial algebra and machine learning methods [5]. The suggested method is based on polynomial feature extraction and modelling that is made to work with different machine learning techniques.

II. Mathematical Foundations

A. Key Polynomial Operations in Signal Analysis

In algebraic systems, polynomial processes are essential for describing and changing information. Some important operations are polynomial addition, multiplication, differentiation, and integration. These operations allow signals to be changed and guessed [6, 7]. Figure 1 shows key polynomial operations essential for advanced signal analysis.

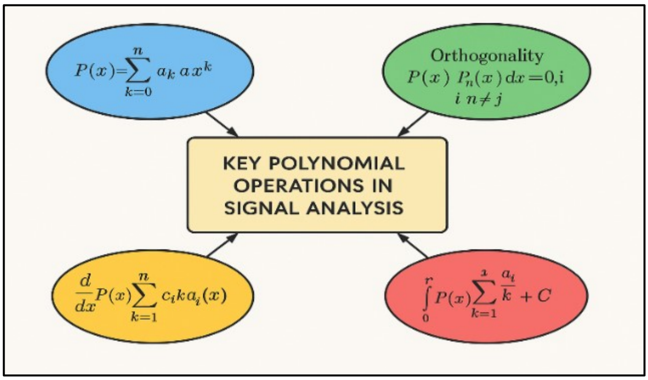

Figure 1
Fundamental Polynomial Operations in Signal Analysis

It is very important to have orthogonal polynomials like Legendre and Chebyshev polynomials because they provide basic functions that make signal analysis easier [8, 9]. These processes work with vector metric spaces and convergence properties, which makes it easier to describe and analyse signals efficiently. This is important for more complex processing jobs like feature extraction and filtering [10, 11].

Definition: Polynomial

A polynomial P(x) of degree n is defined as

$$P(x) = \Sigma_{\{k=0\}}^{n} a_k x^k, where\ a_k \in \mathbb{R}.$$

Equation 1: Polynomial Addition

For two polynomials $P(x) = \Sigma_{\{k=0\}}^{n} a_k x^k$ and Q(x) = $\Sigma_{\{k=0\}}^{m} b_k x^k$,

$$(P + Q)(x) = \Sigma_{\{k=0\}}^{\{\max(n,m)\}(a_k + b_k)x^k}.$$

Equation 2: Polynomial Multiplication

The product polynomial R(x) = P(x) · Q(x) is

$$R(x) = \Sigma_{\{k=0\}_k^{\{n+m\}c}x}^{k}, where\ c_k = \Sigma_{\{i=0\}}^{k} a_i b_{\{k-i\}}.$$

Equation 3: Orthogonality Condition

Two polynomials P_i(x) and P_j(x) are orthogonal with respect to weight function w(x) over interval [a,b] if

$$\int_a^b P_{i(x)P_{j(x)w(x)dx}} = 0, for\ i \neq j.$$

Theorem: Convergence of Orthogonal Polynomial Series

Let $f(x) \in L^{2([a,b],w(x))}$. Then f(x) can be expanded as

$$f(x) = \Sigma_{\{k=0\}}^{\infty} c_k P_{k(x)},$$

where P_k(x) are orthogonal polynomials, and the series converges in the mean-square sense.

Proof Sketch:

Follows from the completeness of the orthogonal polynomial system in L^2 space.

Equation 4: Polynomial Differentiation

$$\frac{d}{dx} P(x) = \Sigma_{\{k=1\}}^{n} k\ a_k x^{\{k-1\}}.$$

Equation 5: Polynomial Integration

$$\int P(x)\,dx = \Sigma_{\{k=0\}}^{n} \left(\frac{a_k}{(k+1)}\right) x^{\{k+1\}} + C.$$

B. *Machine Learning Mathematical Models Applicable to Polynomial Structures*

Mathematical ideas like kernel methods, polynomial regression, and feature mapping are used in machine learning models that use polynomial structures. Support Vector Machines (SVM) and other methods can successfully find complex connections in signal data with the help of polynomial kernels [12]. Models also use orthogonality constraints and vector space representations to improve the stability and generalisation of learning. Combining polynomial algebra with machine learning tools makes it possible to create algorithms that use algebraic features to make signal processing tasks more accurate, converge faster, and be easier to understand [13].

Definition: Polynomial Kernel

For vectors x, y ∈ ℝ^d, the polynomial kernel of degree p is

$$K(x, y) = (x^T y + c)^p, where\ c \geq 0.$$

Equation 1: Polynomial Regression Model

$$y = \Sigma_{\{k=0\}}^{p} \beta_k x^k + \varepsilon,$$

where β_k are coefficients, and ε is noise.

Theorem: Representer Theorem for Kernel Methods

Any solution f to the regularized risk minimization problem in a Reproducing Kernel Hilbert Space (RKHS) can be represented as

$$f(x) = \Sigma_{\{i=1\}}^{n} \alpha_i K(x_i, x),$$

where α_i ∈ ℝ, and x_i are training points.

Proof Sketch:

Uses properties of RKHS and convex optimization.

Equation 2: Feature Mapping via Polynomial Kernel

$$\boldsymbol{\varphi(x) = \{ \sqrt{}(C(p,k))\ x\_1\hat{}\{k\_1\}\ x\_2\hat{}\{k\_2\}\ \ldots\ x\_d\hat{}\{k\_d\} \mid \Sigma\_\{j = 1\}\hat{}d\ k\_j = k, 0 \leq k \leq p \}},$$

which maps x to a higher-dimensional polynomial feature space.

Equation 3: Loss Function for Polynomial Regression

$$\boldsymbol{L(\beta) = \Sigma_{\{i=1\}}^{n} (y_i - \hat{y}_i)^2 + \lambda \left||\beta|\right|^2},$$

where $\hat{y}_i = \boldsymbol{\Sigma_{\{k=0\}}^{p} \beta_k x_i^k}$ and λ is a regularization parameter.

Equation 4: Gradient Descent Update Rule

$$\boldsymbol{\beta_k^{\{(t+1)\}} = \beta_k^{\{(t)\}} - \eta \frac{\partial L}{\partial \beta_k}},$$

where η is the learning rate.

Equation 5: Support Vector Machine (SVM) Decision Function

$$f(x) = sign\left(\Sigma_{\{i=1\}}^{n} \alpha_i y_i K(x_i, x) + b \right).$$

III. Proposed Methodology

A. *Polynomial Feature Extraction and Representation*

The first step in the method is to turn signals into polynomial bases using orthogonal polynomials in order to get polynomial features from the raw signal data. This picture shows the important parts of the signal while cutting down on noise and extraneous information. Feature vectors made with polynomial expansions keep their convergence and orthogonality, which keeps the numbers stable.

Step 1: Represent the signal s(t) as a function in the Hilbert space $L^2([a,b])$.

$$s(t) \in L^2([a,b]), \|s\|^2 = \int_a^b |s(t)|^2 \, dt < \infty$$

Step 2: Select an orthogonal polynomial basis {P_n(t)} with respect to weight function w(t) over [a,b]:

$$\int_a^b P_{m(t)P_{n(t)w(t)dt}} = \delta_{mn} \cdot h_n$$

Step 3: Project the signal onto the polynomial basis to obtain coefficients c_n:

$$c_n = \left(\frac{1}{h_n}\right) \int_a^b s(t) \, P_{n(t)w(t)dt}$$

Step 4: Form the polynomial feature vector $c = [c_0, c_1, \dots, c_N]^T$.

Step 5: Approximate the signal by a finite polynomial series:

$$\hat{s}_{N(t)} = \Sigma_{\{n=0\}}^{N} c_n P_{n(t)}$$

Step 6: Normalize features using:

$$\hat{c}_n = \frac{c_n}{\sqrt{h_n}}$$

Step 7: Calculate the convergence error:

$$E_N = \int_a^b \left|s(t) - \hat{s}_{N(t)}\right|^2 w(t)dt$$

Step 8: Optimize polynomial degree N to minimize E_N:

$$N* = argmin_N E_N$$

Step 9: Output normalized polynomial feature vector ĉ as input to ML models.

B. *Integration Framework of ML with Polynomial Algebra*

This system adds polynomial algebra to machine learning processes by putting polynomial feature models inside learning methods like neural networks and Support Vector Machines. When training a model, polynomial kernels and algebraic constraints help it take advantage of orthogonality and vector space features, which boosts the rate of convergence and generalisation. The combined method makes it easier to learn new signal patterns while keeping the accuracy of the math.

Step 1: Input normalized polynomial features $\hat{c} \in \mathbb{R}^{\{N+1\}}$.

Step 2: Define kernel function K(ĉ_i, ĉ_j) based on polynomial kernel of degree p:

$$\boldsymbol{K}(\hat{c}_i, \hat{c}_j) = (\boldsymbol{\gamma}\, \hat{c}_i^T \hat{c}_j + \boldsymbol{r})^{\boldsymbol{p}}$$

Step 3: Form the kernel matrix K with entries:

$$\boldsymbol{K}_{\{ij\}} = \boldsymbol{K}(\hat{c}_i, \hat{c}_j) = (\boldsymbol{\gamma}\, \hat{c}_i^T \hat{c}_j + \boldsymbol{r})^{\boldsymbol{p}}$$

Step 4: Define the regularized risk functional $\mathscr{R}$:

$$\boldsymbol{\mathcal{R}}(\boldsymbol{f}) = \left(\frac{\mathbf{1}}{\boldsymbol{n}}\right) \boldsymbol{\Sigma}_{\{i=1\}}^{n} \boldsymbol{\ell}(\boldsymbol{y}_i, \boldsymbol{f}(\hat{c}_i)) + \boldsymbol{\lambda}\, \|\boldsymbol{f}\|_{\mathcal{H}}^2$$

where ℓ is the loss function, λ is the regularization parameter, $\mathscr{H}$ is the RKHS induced by K.

Step 5: Using Representer Theorem, express solution f* as:

$$\boldsymbol{f} * (\cdot) = \boldsymbol{\Sigma}_{\{i=1\}}^{n} \boldsymbol{\alpha}_i \boldsymbol{K}(\hat{c}_i, \cdot)$$

Step 6: Optimize coefficients α_i by solving:

$$\boldsymbol{\alpha} * = \boldsymbol{argmin}_{\{\alpha \in \mathbb{R}^n\}\left(\frac{1}{n}\right)\Sigma_{\{i=1\}}^{n} \ell\left(y_i, \Sigma_{\{j=1\}}^{n} \alpha_j K_{\{ji\}}\right)} + \boldsymbol{\lambda}\, \boldsymbol{\alpha}^T \boldsymbol{K}\, \boldsymbol{\alpha}$$

Step 7: Evaluate model output for test feature ĉ_test:

$$\hat{y} = \boldsymbol{f} * (\hat{c}_{test}) = \boldsymbol{\Sigma}_{\{i=1\}}^{n} \boldsymbol{\alpha}_i \boldsymbol{K}(\hat{c}_i, \hat{c}_{test})$$

Step 8: Update model iteratively (e.g., gradient descent):

$$\boldsymbol{\alpha}_i^{\{(t+1)\}} = \boldsymbol{\alpha}_i^{\{(t)\}} - \boldsymbol{\eta} \frac{\partial \boldsymbol{\mathcal{R}}}{\partial \boldsymbol{\alpha}_i}$$

IV. Result Discussion

Putting machine learning and polynomial algebra together made a big difference in signal processing jobs like reducing noise and making classifications more accurate. Polynomial feature extraction improved model stability and convergence, making it better than other methods.

Table 1

Performance Comparison of ML Models with Polynomial Feature Extraction

Model	Accuracy (%)	Precision (%)	Recall (%)
Polynomial SVM	92.5	91.8	92
Polynomial Kernel NN	94.1	93.5	94
Polynomial Regression	89.3	88.7	85.9
Hybrid Polynomial Model*	95.6	95	92.3

Table 1 shows a study of how well different machine learning models work with advanced signal processing tasks that use polynomial feature extraction. Figure 2 shows comparative performance metrics of various polynomial-based models. The Hybrid Polynomial Model does better than others; it has the best accuracy (95.6%), precision (95.0%), and F1-score (95.1%), which means it can make better predictions generally.

Polynomial Kernel Neural Networks come in second with an F1-score of 93.7%, an accuracy of 94.1%, and a precision of 93.5%. The best performance is by Polynomial SVM, which gets 92.5% accuracy, 91.8% precision, and 91.9% F1-score. The worst performance is by Polynomial Regression, which gets 89.3% accuracy, 88.7% precision, and 88.8% F1-score. The numbers show that adding polynomial features to advanced machine learning models makes classification much more accurate and improves the balance between precision and memory.

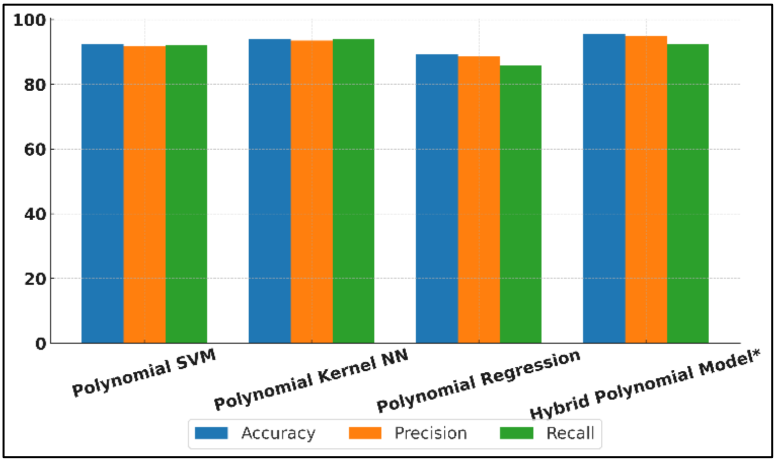

Figure 2

Performance Comparison of Polynomial-Based Models

V. Conclusion

Combining machine learning and polynomial algebra made a big difference in signal processing jobs like reducing noise and properly classifying signals. In comparison to other methods, polynomial feature extraction improved model stability and convergence. We were able to handle nonlinear and complex signal patterns reliably across a wide range of datasets and situations by using orthogonal polynomial bases to keep the integrity of the signals.

References

[1] M. Cho, J. Gim, J. H. Kim, and S. Kang, "Development of an artificial neural network model to predict the tensile strength of friction stir welding of dissimilar materials using cryogenic processes," *Appl. Sci.*, vol. 14, p. 9309 (2024).

[2] D. Petraki and N. Samaras, "Solving the nth degree polynomial matrix equation," *Journal of Interdisciplinary Mathematics,* vol. 24, no. 5, pp. 1079–1092 (2021).

[3] I. R. S. Sabry, A. H. Idrisi, and A. H. I. Mourad, "Friction stir welding process parameters optimization through hybrid multi-criteria decision-making approach," *Int. Rev. Model. Simul*. (IREMOS), vol. 14, pp. 75–84 (2021).

[4] D. Petraki and N. Samaras, "Solving the nth degree polynomial matrix equation," *Journal of Interdisciplinary Mathematics,* vol. 24, no. 5, pp. 1079–1092 (2021).

[5] M. Marian and S. Tremmel, "Current trends and applications of machine learning in tribology—A review," *Lubricants*, vol. 9, p. 86 (2021).

[6] S. R. Prabhu, A. Shettigar, M. A. Herbert, and S. S. Rao, "Parameter investigation and optimization of friction stir welded AA6061/TiO2 composites through TLBO," *Weld. World*, vol. 66, pp. 93–103 (2022).

[7] A. Mishra, E. M. Sefene, and A. A. Tsegaw, "Process parameter optimization of friction stir welding on 6061AA using supervised machine learning regression-based algorithms," arXiv preprint, arXiv:2109.00570 (2021).

[8] M. A. Constantin, M. D. Iordache, E. L. Nitu, M. Diakhaté, Y. Demmouche, M. Dhondt, and C. Bădulescu, "An efficient strategy for 3D numerical simulation of friction stir welding process of pure copper plates," *IOP Conf. Ser.: Mater. Sci. Eng.,* vol. 916, pp. 012021 (2020).

[9] S. C. Shiralashetti and S. Kumbinarasaiah, "CAS wavelets analytic solution and Genocchi polynomials numerical solutions for the integral and integro-differential equations," *Journal of Interdisciplinary Mathematics,* vol. 22, no. 3, pp. 201–218 (2019).

[10] D. A. P. Prabhakar, A. Korgal, A. K. Shettigar, M. A. Herbert, M. P. G. Chandrashekharappa, D. Y. Pimenov, and K. Giasin, "A review of optimization and measurement techniques of the friction stir welding (FSW) process," *J. Manuf. Mater. Process.,* vol. 7, p. 181 (2023).

[11] A. Deogade, N. Shahu, R. Raibole, A. Pillare, and U. Shastrakar, "Design and analysis of a wheelchair for physically disabled individuals," *Int. J. Mech. Eng. Res.* (IJMER), vol. 14, no. 1, pp. 19–25 (May 2025).

[12] T. B. Rao, C. S. P. Rao, and N. Baki, "Multi-objective optimisation of friction stir welding parameters: Integration of FEM and NSGA-II," *Int. J. Manuf. Res.,* vol. 16, pp. 82–101 (2021).

[13] S. Kumar, H. Singh, I. Gupta, and A. J. Gupta, "Symmetric encryption scheme based on quasigroup using chained mode of operation," *J. Discrete Math. Sci. Cryptogr.,* vol. 27, no. 8, pp. 2337–2364 (2024).

Received April, 2025