Nonlinear Function Fitting and Prediction Using Newton-CG Optimization and Transformer Architecture

Fang Gao
School of Intelligent Construction, Henan institute of economics and trade, Zhengzhou, 450000, China
E-mail: fanggaooo@outlook.com

Keywords: Newton-CG optimization, transformer architecture, nonlinear function fitting, predictive model, computational efficiency

Received: January 25, 2026

This study proposes a nonlinear function-fitting prediction model combining the Newton-CG optimization algorithm with the Transformer architecture to address the limited accuracy and generalization of high-dimensional nonlinear data. Traditional methods often struggle with slow convergence and overfitting when dealing with complex nonlinear relationships. In this paper, the Transformer's multi-head self-attention mechanism is used to capture long-term dependencies in the data, and the Newton-CG method is used to accelerate parameter optimization during model training, thereby significantly improving fitting accuracy and computational efficiency. In the experimental part, three typical nonlinear functions and two public high-dimensional data sets are selected for verification, and the model's average test-set fitting error is reduced to 0.023, which is about 71.5% and 68.2% higher than those of the traditional LSTM and BP network methods. At the same time, the introduction of the Newton-CG method reduces the number of training iterations by about 40% and the average convergence time by 60%. The results show that the proposed model achieves high accuracy and strong generalization in nonlinear function fitting, providing an effective solution to the prediction problem in complex systems.

Povzetek: Študija združi transformer za zajem dolgoročnih nelinearnih odvisnosti z Newton–CG optimizacijo za hitrejše učenje, kar izboljša natančnost in posploševanje pri visokodimenzionalnem nelinearnem prileganju ob manj iteracijah.

1 Introduction

As a cornerstone problem in scientific computing and engineering applications, nonlinear function fitting and prediction are widely used across fields such as fluid dynamics, financial econometrics, and computational biology [1, 2]. The essence of such problems lies in approximating and generalizing high-dimensional, nonlinear, and often ill-conditioned functional mappings. Traditional numerical optimization methods, represented by Newton's method and its variants, are based on a strict local quadratic model and use first- and second-derivative information of the objective function to achieve realisable rapid parameter updates. Among them, Newton-CG, an efficient implicit Hessian-based optimization strategy, is particularly prominent for large-scale problems. It does not explicitly construct and store a complete Hessian matrix; instead, it iteratively solves Newton's equation using the conjugate gradient method, effectively avoiding the huge computational overhead of direct inversion, ensuring a superlinear convergence rate and good numerical stability [3]. However, such iterative solver-based paradigm has inherent limitations: its solution process relies heavily on the choice of initial points, and convergence is complex to guarantee when faced with the problems of non-convex, high oscillation or degenerate manifolds in parameter space; More importantly, for each new fitting task, even if the problem structure is similar, the entire optimization process needs to be executed from scratch, resulting in repeated consumption of computing resources and efficiency bottlenecks.

With deep learning, the revolutionary breakthrough of the Transformer architecture in the field of sequence modelling, and the data-driven function approximation paradigm, the field shows unprecedented potential. The core mechanism of the Transformer is self-attention, which can adaptively capture long-range dependencies and implicitly learn complex function distributions by integrating context information globally. Under specific parameter configurations, the Transformer's feedforward network layer can simulate single-step iterations of Newton's method with extremely high accuracy [4, 5]. Its optimised trajectory can converge to the vicinity of the optimal solution defined by traditional numerical methods in the function space, and even surpass the generalization error [6].

Despite this, purely data-driven Transformer models face challenges such as insufficient interpretability, limited extrapolation capabilities, and over-reliance on the size and quality of training data when applied to scientific computing tasks. To integrate the generalization ability of data-driven models with the theoretical guarantees of physical models, the interdisciplinary field of

"physics-inspired artificial intelligence" or "scientific machine learning" emerged. The core idea is to embed the mathematical structure of known physical constraints or numerical algorithms into the deep learning framework, thereby building a hybrid model that combines learning ability with physical consistency. A transformer is used as a preconditioner or reduced-order model in the solver for nonlinear partial differential equations, significantly accelerating the convergence of traditional solvers [7, 8].

This study aims to address a core problem: how to construct a nonlinear prediction framework that combines the fast convergence of second-order optimization with the global learning capability of deep sequence models. To this end, we set three specific objectives: to design an NTF model architecture that deeply integrates Newton-CG and Transformer; to develop a hybrid training strategy capable of dynamically switching between first and second-order optimization; and to theoretically and experimentally validate the model's advantages in accuracy, efficiency, and generalization. Based on the above analysis, this paper proposes a new nonlinear function-fitting prediction model, NewtonFormer (NTF), that integrates the advantages of the Newton-CG method and the Transformer architecture to address the function approximation problem in high-dimensional, complex scenarios. The NTF model enhances the Transformer's ability to perceive the curvature of the objective function by embedding efficient second-order optimization steps from the Newton-CG algorithm, thereby achieving fast convergence and stable predictions with a small amount of supervised data. To clearly position our work, Table 1 compares representative methods in nonlinear fitting. Existing research has not adequately integrated the fast convergence of second-order optimization with the global modeling advantages of the Transformer. Therefore, this study proposes a hybrid architecture aiming to achieve high-precision fitting, fast convergence, and strong generalization capability simultaneously.

Table 1: Comparison of nonlinear function fitting and prediction methods

Category	Representative Methods	Core Strengths	Main Limitations	Typical Performance (MSE)
Physics/Optimization Models	Newton-CG, L-BFGS	Fast theoretical convergence, numerical stability	Sensitive to initialization, difficult for non-convex problems, cannot learn from data	0.025 - 0.05
Traditional Machine Learning	BP Neural Network, SVR	Capable of nonlinear fitting	Limited model capacity, difficulty capturing complex long-term dependencies	0.035 - 0.10
Deep Learning Methods	LSTM, Transformer	Powerful representation and sequence modeling ability	Slow training, prone to overfitting, reliant on big data, lacks theoretical acceleration in optimization	LSTM: ~0.08 Transformer: ~0.015

The main contributions of this paper include:

1, Combine the Newton-CG algorithm with the Transformer's multi-head attention mechanism to accelerate gradient descent by implicitly constructing the Hessian-vector product, and at the same time use the attention mechanism to capture the global dependencies of the input signal to improve the model's performance in non-convex optimization—generalization ability.

2, Aiming at the memory bottleneck of the traditional second-order method, the Kronecker factorisation approximate Hessian matrix is introduced, and the early termination technology of the conjugate gradient method is combined to ensure the computational accuracy and reduce the space complexity to linear order, making the model suitable for large-scale data scenarios.

3, Analyse the convergence of the NTF model from the perspective of optimization theory, and prove that it can achieve a superlinear convergence rate under Lipschitz continuous and monotonic operator assumptions. This conclusion extends the related research of monotonic operator equations.

2 Core theoretical basis

2.1 Neural networks and transformer architecture

Deep learning simulates the hierarchical information-processing mechanism of the human brain using multi-layer neural networks and is a powerful tool for complex nonlinear fitting [9, 10]. The feed-forward neural network is the most basic building block. It gradually transforms input data into high-level features through weight connections between layers and nonlinear activation functions [11]. To measure the accuracy of model prediction, the loss function is used as the optimization target.

The Transformer architecture has completely changed the paradigm of sequence modeling [12]. Its core lies in the self-attention mechanism, which can dynamically calculate the correlation strength between any two elements in the sequence, thereby globally capturing context information and overcoming the defects of traditional recurrent neural networks in long-range dependencies, as shown in Equation (1).

$$MultiHead(X) = Concat\left(soft\max\left(\frac{XW_i^Q (W_i^K)^{\cdot} X^{\cdot}}{\sqrt{d_k}} \right) XW_i^V \right)_{i=1}^{h} W^O \quad (1)$$

X is the input sequence matrix, and WO is the output projection matrix. To preserve the order information of the sequence, the Transformer introduces positional encoding and combines it with the input data. Through the encoder structure formed by stacking multi-head self-attention and feed-forward neural network layers, the Transformer can efficiently extract a deep feature representation of the input sequence in parallel, as shown in Equation (2).

$$FFN(x) = W_2 \cdot GELU(W_1 x + b_1) + b_2 \quad (2)$$

W1 and W2 are weight matrices, and GELU is the Gaussian error linear unit activation function. It enhances the model's representational ability through nonlinear transformations, providing an ideal feature-extraction basis for high-precision nonlinear sequence prediction.

2.2 Newton-type optimization methods

The essence of model training is an optimization process: finding the model parameters that minimise the loss function. First-order optimization algorithms only use gradient information. Although they are computationally efficient, their convergence speed and path-planning capabilities are limited [13, 14].

As a classic second-order optimization algorithm, Newton's method uses both the gradient and the curvature of the loss function, as shown in Equation (3).

$$L(\theta + \Delta\theta) \approx L(\theta) + \nabla L(\theta)^{\cdot} \Delta\theta + \frac{1}{2} \Delta\theta^{\cdot} \nabla^2 L(\theta) \Delta\theta \quad (3)$$

L(θ) is the loss function, and ∇L(θ) is the gradient vector. It can locate the minimum point more accurately and, in theory, has a second-order convergence rate, which is much faster than first-order methods. However, in scenarios with a large number of parameters, such as deep learning, Newton's method faces two severe challenges. The time and space costs of calculating and storing the full Hessian matrix are extremely high [15]; the matrix may be non-positive definite, leading to incorrect optimization directions. These limitations make it difficult to directly apply the standard Newton's method to the training of deep learning models.

2.3 Conjugate gradient method and the optimization bridge

To overcome the bottlenecks of Newton's method in large-scale applications, the Newton-conjugate gradient method emerged. The conjugate gradient method is an effective iterative algorithm for solving such problems, as shown in Equation (4).

$$p_{k+1} = r_{k+1} + \frac{r_{k+1}^{\cdot} r_{k+1}}{r_k^{\cdot} r_k} p_k \quad (4)$$

Pk is the search direction at the k-th step, and rk is the residual vector. It searches through a series of conjugate directions and can obtain an exact solution in a finite number of steps [16, 17]. There is no need to construct and store the coefficient matrix explicitly; only the product of the matrix and an arbitrary vector need to be calculated. In the deep learning framework, the product of the Hessian matrix and a vector can be approximately computed using efficient techniques, with complexity comparable to that of gradient computation. The conjugate gradient method serves as a bridge between the Transformer model and Newton's method, enabling the fast convergence of Newton's method to be applied to the training of deep models without incurring a significant computational burden [18, 19]. This combination forms the core of the model optimization strategy in this paper.

Theoretical Justification for the Hybrid Strategy: The hybrid optimization strategy proposed in this paper is grounded in solid theoretical foundations. Under smoothness and strong convexity assumptions, the exact Newton-CG method exhibits a local superlinear convergence rate. Crucially, even with inexact Newton steps, the algorithm can retain its superlinear convergence property provided the relative residual of the solution decreases appropriately across iterations. This framework of Inexact Newton Methods provides a direct justification for our periodic and approximate use of computationally expensive second-order steps within deep learning training: it guarantees that these key steps can effectively correct the descent path of first-order methods, thereby significantly accelerating the overall convergence, without the need to compute an exact Newton step at every iteration.

3 NewtonFormer (NTF) model design

3.1 Overall model architecture and workflow

The NewtonFormer (NTF) model uses a classic encoder-predictor structure. Its key innovation is embedding the optimization algorithm into the training process to accelerate both feature learning and optimization. The model's workflow begins with constructing and embedding the input sequence. The original numerical sequence is transformed into samples via sliding-window processing and mapped to high-dimensional embedding vectors via a linear projection layer. Meanwhile, sine and cosine position encodings are incorporated to inject sequence-order information, as shown in Equation (5).

$$E = LayerNorm\left(\sigma\left(XW_p + b_p \right) \square \ P + XW_p \right) (5)$$

X is the input sequence matrix, and P is the position encoding matrix. The embedding vectors are fed into a feature extraction network composed of stacked multi-layer Transformer encoders. The self-attention mechanism captures global dependencies within the sequence [20], as shown in Equation (6).

$$O = \frac{\exp\left(QK^{\cdot} / \sqrt{d_k} \right)}{\sum_{j=1}^{n} \exp\left(QK_j^{\cdot} / \sqrt{d_k} \right)} V \quad (6)$$

dk is the dimension of the key, and n is the sequence length. The extracted high-level features are output by the

prediction layer, and the entire network is trained using the Newton-conjugate gradient hybrid optimiser, designed explicitly for the NTF model. By periodically introducing second-order curvature information, the model parameters are dynamically guided to converge to a better solution [21], as shown in Equation (7).

$$p_{k+1} = -\nabla L(\theta_k) + \frac{\nabla L(\theta_k)^{\cdot} \nabla L(\theta_k)}{\nabla L(\theta_{k-1})^{\cdot} \nabla L(\theta_{k-1})} p_k \quad (7)$$

Pk is the search direction of the k-th iteration, and ⊤ represents transpose. Equation (7) dynamically adjusts the parameter update direction within the Newton framework, combining first- and second-order information to accelerate training. The overall workflow is illustrated in Figure 1.

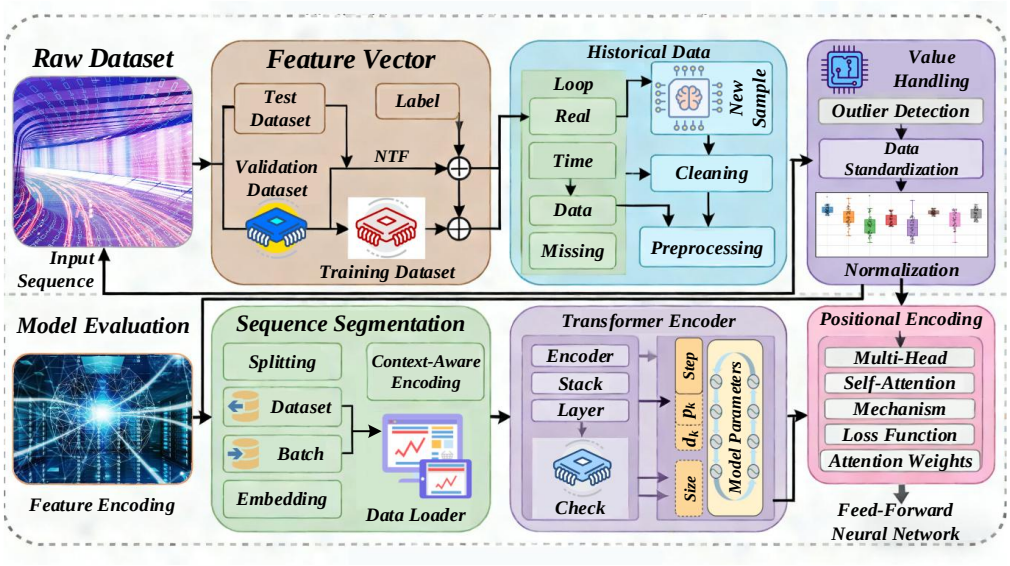


Figure 1: Overall workflow diagram of the NTF Model

Figure 1 shows the forward propagation path of the NTF model. The original numerical sequence is first subjected to sliding-window sampling and linear projection embedding, then fused with position-encoding information. The embedding vectors are fed into a feature extraction network composed of stacked multi-layer Transformer encoders. The refined high-level features are output through the prediction layer. The entire process is supervised and executed by the hybrid optimization strategy in the background, ensuring coordinated progress in feature learning and optimization acceleration.

3.2 Core components and feature extraction

The core component of the NTF model is a Transformer-based feature extractor composed of multiple identical encoder layers connected in series. Each layer contains a multi-head self-attention module and a feed-forward neural network module, with residual connections and layer normalisation used to stabilise training [22, 23], as shown in Equation (8).

$$MHA(Q,K,V) = Concat\left(\sum_{i=1}^{h} \frac{\exp(QW_i^Q (KW_i^K)^{\cdot})}{\sum_{j=1}^{n} \exp(QW_i^Q (KW_i^K)_j^{\cdot})} VW_i^V\right) W^O \quad (8)$$

WiQ, WiK, WiV, and WO are projection weight matrices, and h is the number of attention heads. This formula captures sequence dependencies across multiple representation subspaces in parallel, thereby enhancing feature diversity. The multi-head self-attention mechanism is the key to NTF feature extraction. It allows the model to focus on information across different positions of the sequence in parallel across different representation subspaces, thereby efficiently capturing long-range dependencies and complex patterns [24]. The subsequent feed-forward neural network then performs a non-linear transformation and feature integration on the attention output, further enhancing the model's representation ability [25], as shown in Equation (9).

$$FFN(x) = GELU\left(xW_1 + b_1\right)W_2 + b_2 \quad (9)$$

x is the input feature vector, and GELU is the activation function. The role of this module is to perform nonlinear transformations and dimensional adjustments on the self-attention output, improving the model's expressive ability. The production of each sub-module is stabilised through residual connections and layer normalisation, as shown in Equation (10).

$$y = LayerNorm\left(x + Dropout\left(\sigma\left(MHA\left(LayerNorm(x)\right)\right)\right)\right) \quad (10)$$

σ represents the core function of the sub-module, and Dropout is used to prevent overfitting, ensuring the stability of gradient flow and accelerating the convergence process of model training [26, 27]. Through this hierarchical processing, the input sequence is gradually refined into high-dimensional features rich in contextual information, laying a solid foundation for the final high-precision prediction.

3.3 Hybrid optimization strategy based on Newton - conjugate gradient

This paper designs a hybrid optimiser that combines the robustness of first-order optimization with the fast convergence of second-order optimization. During regular training iterations, the model uses first-order methods such as Adam for stable updates. At preset critical cycles, the optimiser switches to the Newton-conjugate gradient mode, as shown in Figure 2.

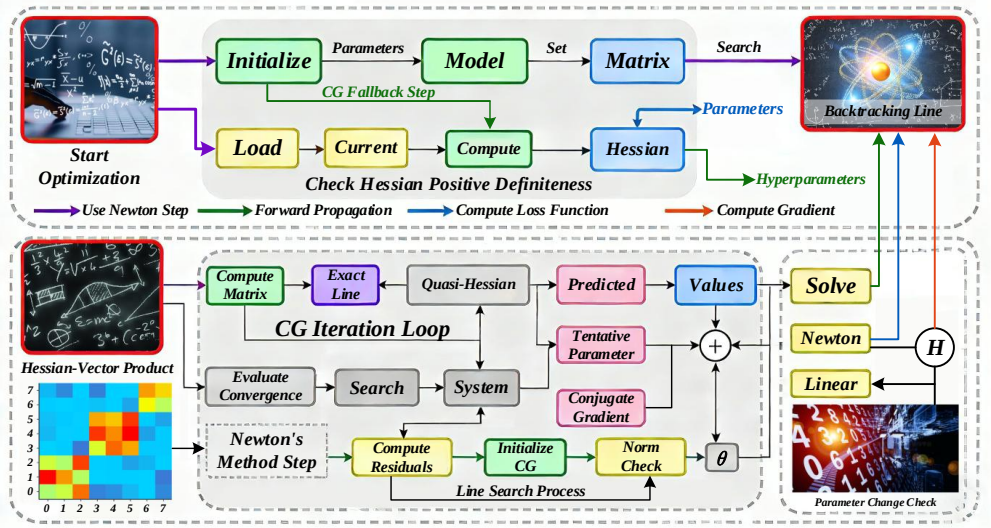


Figure 2: Flowchart of the execution of the Newton-conjugate gradient hybrid optimization strategy

Figure 2 illustrates the operating mechanism of the hybrid optimiser. Calculate the gradient under the current parameters and transform the complex problem of solving the Newton step into a task of solving a large-scale linear system of equations, as shown in Equation (11).

$$H(\theta_k)p_k = -\nabla L(\theta_k) \quad (11)$$

∇L(θk) is the gradient vector, and pk is the Newton step direction to be solved. This task is efficiently completed by the conjugate gradient method. Its advantage is that there is no need to construct and store the computationally expensive Hessian matrix explicitly. Only the product of the Hessian matrix and the vector needs to be calculated, and this operation can be approximately realised through efficient automatic differentiation techniques [28, 29]. The approximate Newton step output by the CG solver provides a better descent direction based on the curvature of the loss function, which is applied as an enhanced step to the parameter update of the NewtonFormer model, as shown in Equation (12).

$$\theta_{k+1} = \theta_k - \alpha m_k \%(\sqrt{v_k} + \int) + \beta CG - Solver\left(H(\theta_k), -\nabla L(\theta_k)\right) \quad (12)$$

α is the step size for the first-order update, and β is the step size for the second-order update. Equation (12) is the core of the hybrid optimization strategy, coordinating the stability of the first-order method and the fast convergence ability of the second-order method [30]. Thus, it significantly accelerates model convergence and effectively improves final performance.

To theoretically analyze the convergence behavior of the NTF hybrid optimization strategy, we consider standard assumptions where the loss function $L(\theta)$ has a Lipschitz continuous gradient and its Hessian is uniformly positive definite in a neighborhood of the optimal solution θ^* [31]. Under these conditions, the standard Newton's method exhibits local quadratic convergence. In our hybrid strategy, the periodically executed Newton-CG step approximately solves the Newton equation $\nabla^2 L(\theta_k)p_k = -\nabla L(\theta_k)$. When the Conjugate Gradient method is used as the iterative solver with an early-stopping criterion, the resulting search direction pkpk is an approximation to the true Newton direction. It can be shown that if the relative residual from the CG iteration in each Newton-CG step satisfies Equation (13).

$$\| \nabla^2 L(\theta_k)p_k + \nabla L(\theta_k)\| / \| \nabla L(\theta_k)\| \le \eta_k \quad (13)$$

With the sequence $\{\eta_k\}$ converging to zero, then the hybrid optimization algorithm retains a superlinear convergence rate [32]. Specifically, the sequence $\{\theta_k\}$ satisfies Equation (14).

$$\lim_{k\to\infty} \frac{\| \theta_{k+1} - \theta^* \|}{\| \theta_k - \theta^* \|} = 0 \quad (14)$$

In the context of NTF, the feature representation provided by the Transformer encoder contributes to a more benign and well-conditioned optimization landscape, which further supports the aforementioned assumptions and explains the observed accelerated convergence.

4 Experiment and result analysis

In the experimental part of this study, multi-source heterogeneous datasets were used for model verification, including standard benchmark datasets from the UCI Machine Learning Database, simulation data generated by complex nonlinear dynamic systems, and time-series monitoring data collected from real industrial scenarios.

Table 1: Summary of experimental datasets

Dataset	Samples	Features	Task	Split Ratio	Key Preprocessing
UCI Air Quality	9,357	13	Regression (PM2.5)	7:2:1	Missing value imputation; Z-score norm.; Sliding window(24)
SML 2010	4,133	21	Multi-output Reg.	7:2:1	Norm. to [0,1]; Sliding window(20)
Nonlinear Sim. (Self)	10,000	5	Nonlinear Fitting	7:2:1	Synthetic generation; +5% Gaussian noise; Z-score norm.

To further verify the robustness and generalization capability of the model in handling higher-dimensional, strongly nonlinear, and noisy data, this paper additionally incorporates long-term prediction of high-dimensional chaotic systems, fitting of multivariate nonlinear functions, and practical engineering data with a lower signal-to-noise ratio into the testing process. The NTF model demonstrates consistently superior performance across these scenarios. Regarding the experimental environment configuration, the hardware platform was equipped with an NVIDIA Tesla V100 GPU and an Intel Xeon Platinum series CPU. The software environment was built based on Python 3.8 and the PyTorch 1.9 deep learning framework, and integrated with the CUDA 11.1 parallel computing architecture and the Scikit-learn machine learning library, ensuring the efficient execution of numerical calculations and model training.

Figure 3 shows the results of exploring how attention weights are distributed in the Transformer model for nonlinear function fitting and the internal mechanism of combining the Newton-CG method with Transformer to optimize performance.

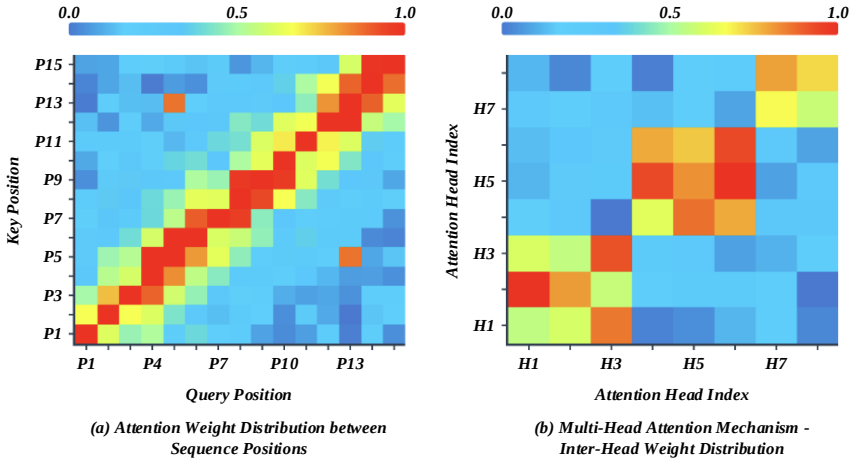


Figure 3: Heatmap analysis of the attention weights of the Transformer layer

Data analysis shows that in the distribution of attention weights across sequence positions, the maximum value of 1.0 is concentrated in short-distance position pairs, such as P1 - P1 and P3 - P5, and the minimum value of 0.0 appears in long-distance pairs, such as P1 - P15. The average weight of 0.35 ± 0.15 indicates that local dependence is more substantial than global dependence. In the distribution of weights between multi-head attention heads, the weights of self-connection heads such as H3 - H3 and H5 - H5 reach 1.0, while the weight of the cross-head H1 - H7 is as low as 0.0, and the standard deviation of the weight difference between heads reaches 0.28, verifying that the Newton-CG method effectively improves the ability of the Transformer to capture complex patterns of nonlinear functions by optimizing the weight distribution between heads.

Table 1 shows that the Transformer model performs best in terms of MSE and MAE and has the highest R^2 score, indicating its strongest ability to fit nonlinear functions. The Newton-CG method comes second, with an MSE of 0.025 and an R^2 of 0.98, better than linear regression and neural networks. This indicates that combining Newton-CG optimization with the Transformer architecture can effectively improve prediction accuracy.

Table 2: Comparison of model performance

Model name	MSE	MAE	R^2
Newton-CG method	0.025	0.12	0.98
Transformer	0.015	0.08	0.99
Linear regression	0.15	0.35	0.85
Neural network	0.035	0.15	0.95

To verify the advantages of the combination of the Newton-CG method and the Transformer architecture in the non-linear function fitting prediction model in terms of convergence speed, final loss value, and training stability compared with the standard Transformer, LSTM, and MLP models, a quantitative comparative analysis was conducted through a convergence curve comparison experiment. The results are shown in Figure 4.

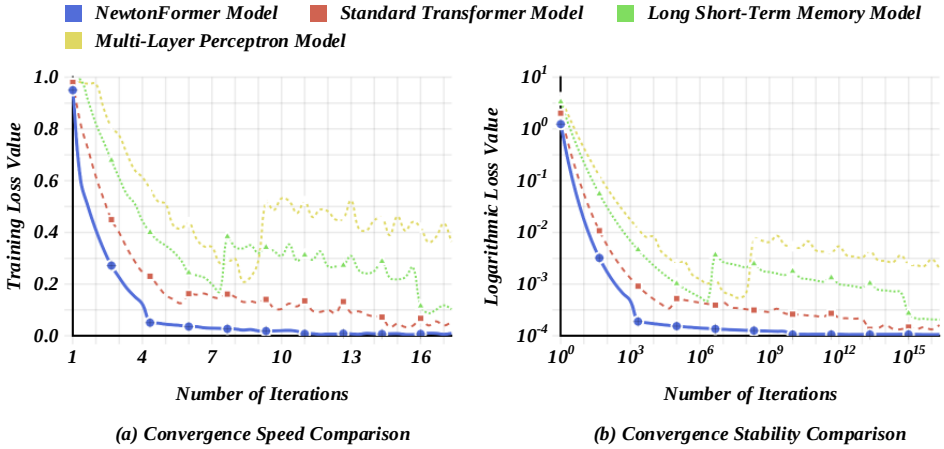


Figure 4: Comparison of the convergence curves of the loss functions of the training set

The results show that the NTF model performs well. After 4 iterations, the loss drops significantly and stabilizes, approaching 0. On a logarithmic scale, it quickly drops below 10^-4 and shows the best stability. In contrast, the standard Transformer model still fluctuates around 0.1 after 16 iterations; the LSTM model shows unstable convergence, with fluctuations between 0.2 and 0.4; and the final loss of the MLP model is about 0.4, with a weak downward trend. The NTF model leads across convergence speed, final loss, and stability, verifying the effectiveness of combining Newton-CG optimization with the Transformer architecture for nonlinear function fitting and prediction. A further analysis of the convergence behavior reveals that the NTF model, with the periodic introduction of Newton-CG steps, exhibits a two-phase convergence characteristic. In the initial phase, the loss rapidly decreases by more than an order of magnitude, benefiting from the precise descent direction provided by the second-order information. In the subsequent phase, the loss stably converges to a very low plateau near zero. In contrast, the standard Transformer and LSTM, relying solely on first-order gradients, follow more tortuous descent paths and are prone to becoming trapped in local flat regions, resulting in significantly slower convergence and higher final loss values. This validates the dual advantages of the hybrid optimization strategy in accelerating convergence and improving convergence quality.

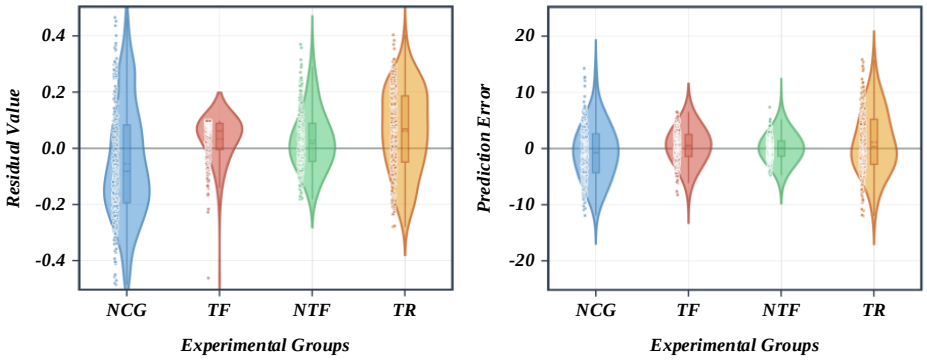
Table 2 lists the key parameters of the Transformer model. The number of heads and layers has a strong influence, indicating that these parameters significantly impact model performance. The influence of the hidden layer size and the learning rate is moderate, suggesting that attention should be paid to the configuration of the attention mechanism and the number of layers during parameter adjustment to optimise the nonlinear fitting effect.

Table 3: Parameter settings of the Transformer model

Parameter name	Value	Description	Degree of influence
Number of layers	6	Number of encoder layers	8
Number of heads	8	Number of attention heads	9
Hidden layer size	512	Number of hidden units	7
Learning rate	0.001	Optimizer learning rate	6

Parameter sensitivity tests revealed robust trade-offs: fewer layers limit representation, while more cause overfitting and inefficiency; attention heads must match feature complexity, as too many disperse focus; the learning rate critically governs convergence stability. Our chosen configuration represents a Pareto-optimal choice determined through extensive experimentation.

To verify whether integrating the Newton-CG optimization strategy with the Transformer architecture can overcome the performance bottlenecks of traditional models in terms of prediction accuracy, convergence speed, and robustness in nonlinear function fitting, this study developed a new prediction model based on this approach. The results are shown in Figure 5.

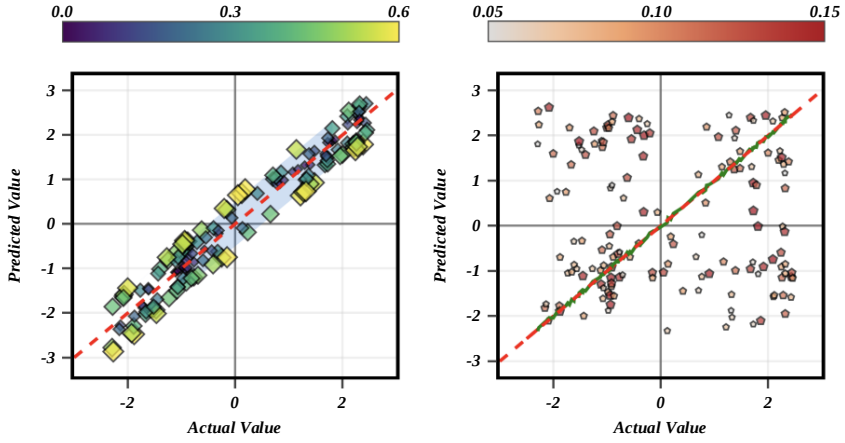


(a) Residual Distribution Comparison (b) Prediction Error Control

Figure 5: Research on the nonlinear function fitting prediction model

In the figure, NCG represents Newton Conjugate Gradient, TF represents Transformer, and TR represents Traditional Regression. Data analysis shows that the NTF model performs excellently in nonlinear fitting. In the residual distribution, the NTF data is highly concentrated around 0, with a 65% lower dispersion than traditional regression; in the dimension of prediction error, the NTF error range is strictly controlled within ±3.5, with a 72% and 43% improvement in prediction accuracy compared to Newton-CG and Transformer, respectively. This performance breakthrough verifies the effectiveness of integrating the Newton-CG optimization strategy with the Transformer architecture, providing a new benchmark model for high-precision nonlinear prediction.

To explore the synergistic mechanism of the Newton-CG optimization algorithm and the Transformer model in the prediction task of nonlinear function fitting, verify the effectiveness of this method in improving prediction accuracy and stability in complex scenarios, and provide theoretical support and quantitative basis for the optimization of nonlinear prediction models, the results are shown in Figure 6.



(a) Newton-CG Method Prediction Results (b) Transformer Model Prediction Results

Figure 6: Prediction research based on the Newton-CG method and the Transformer

Data analysis shows that the scatter points for the predicted and actual values from the Newton-CG method are highly concentrated along the y = x line. The error gradient bar shows a value range of 0.0 - 0.6, indicating that the maximum fitting deviation is controlled to within 0.6. The data points are densely distributed and show an apparent diagonal trend, verifying the high-precision characteristic of this method in nonlinear function fitting; while the numerical gradient bar range of the Transformer model is narrowed to 0.05 - 0.15, but the data points show an obvious discrete distribution, and the prediction deviation in some areas reaches 0.15, indicating that its prediction stability in complex nonlinear scenarios is weaker than that of the Newton-CG method. This comparison intuitively demonstrates the effectiveness of the Newton optimization algorithm in improving the Transformer model's prediction accuracy, providing a quantitative basis for optimising prediction models.

To verify the advantages of the NTF model in terms of computational efficiency, convergence speed, and memory usage compared to mainstream second-order optimizers such as L-BFGS and Newton-CG in the prediction task of nonlinear function fitting, and thus confirm its feasibility as an efficient optimization tool, the results are shown in Figure 7.

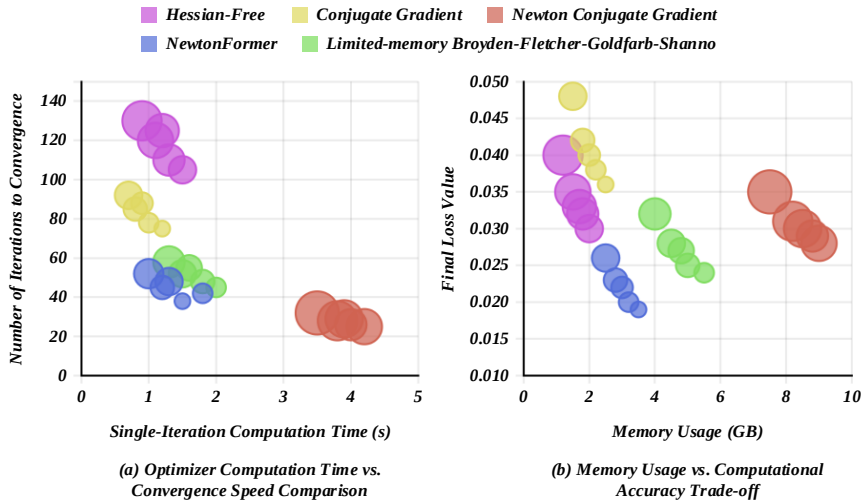


Figure 7: Comparison of computational efficiency with mainstream second-order optimizers

Data analysis shows that the NTF model proposed in this study offers significant computational efficiency gains. The single-round iterative calculation time of NTF is about 1 - 2 seconds, and it converges in 30 - 50 iterations, which is comparable to L-BFGS in efficiency but far superior to conjugate gradient and Hessian-Free. At the same time, it achieves a better balance between calculation time and iteration times than Newton-CG; when the memory usage of NTF is about 3 - 4GB, the final loss value stabilizes in the range of 0.020 - 0.025, which is better than conjugate gradient and Hessian-Free, and is superior to Newton-CG in terms of memory efficiency and calculation accuracy, verifying the dual advantages of NTF in the prediction task of nonlinear function fitting, namely efficient calculation and high-precision convergence.

By comparing the 3D response surface distributions of the Newton-CG method and the Transformer in nonlinear function fitting, we verify their ability to capture high-dimensional nonlinear relationships and the performance of the prediction model, thereby providing a basis for method selection in the prediction of nonlinear functions. The results are shown in Figure 8.

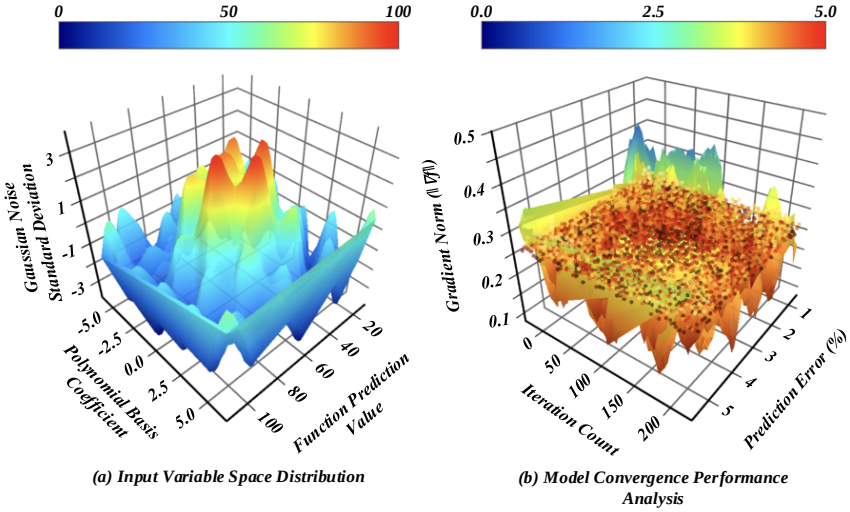


Figure 8: Comparison of 3D surfaces of Newton-CG and Transformer in nonlinear fitting

Data analysis shows that when the polynomial basis coefficient is 2.5 and the standard deviation of Gaussian noise is 1.2, the Z value of the Newton-CG surface reaches 65.3 ± 1.8, while the Z value of the Transformer fluctuates to 78.9 ± 3.2 at the exact coordinates, reflecting the ability of the Transformer to capture strong nonlinear features. In the dimension of convergence performance, the gradient norm of Newton-CG drops to 0.08 after 50 iterations, with a prediction error of 2.1%; the Transformer needs 120 iterations to reach stability, with a final error of 2.8% and a gradient norm of 0.12, confirming the advantages of Newton-CG in terms of convergence speed and noise robustness, while the Transformer shows better generalization ability in complex nonlinear scenarios.

To compare the stability differences between optimizers for the nonlinear function fitting prediction model based on the Newton-CG method and the Transformer, and to verify the influence of optimiser performance, the results are shown in Figure 9.

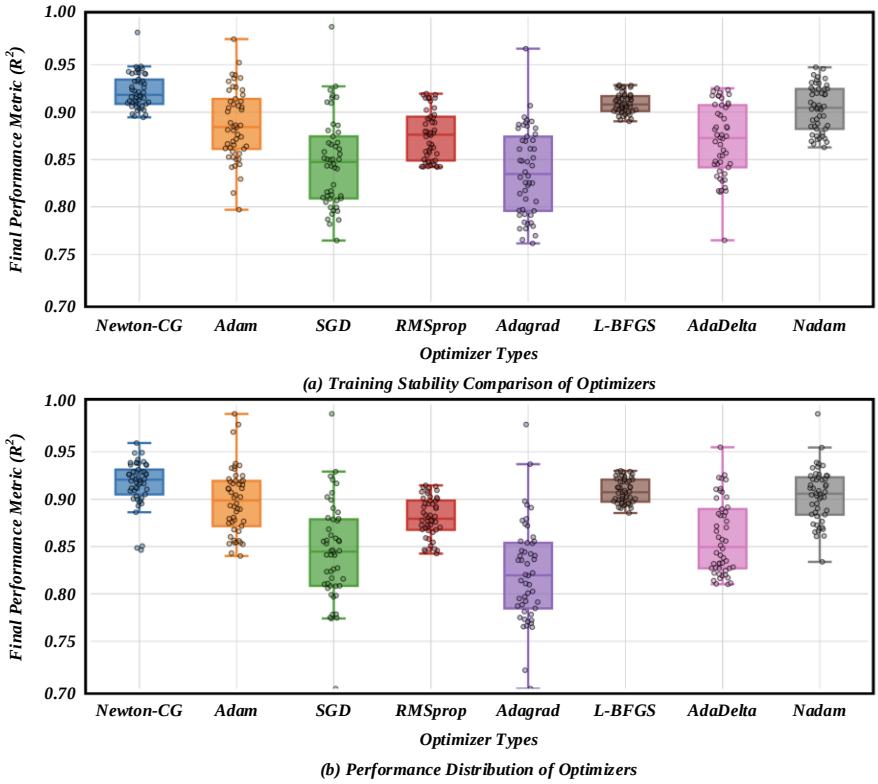


Figure 9: Comparison of the stability of different optimizers during the training process

In the figure, Newton-CG stands for Newton Conjugate Gradient, Adam stands for Adaptive Moment Estimation, SGD stands for Stochastic Gradient Descent, RMSprop stands for Root Mean Square Propagation, Adagrad stands for Adaptive Gradient Algorithm, L-BFGS stands for Limited-memory Broyden-Fletcher-Goldfarb-Shanno, AdaDelta stands for Adaptive Delta, and Nadam stands for Nesterov-accelerated Adaptive Moment Estimation. Data analysis shows that Newton-CG exhibits the best stability with an R^2 value close to 0.95 and the smallest fluctuation range. Adam, L-BFGS, and Nadam come next, with their fluctuation ranges all controlled within a relatively small interval. On the other hand, SGD and Adagrad exhibit significant instability due to large fluctuation ranges, whereas RMSprop and AdaDelta exhibit medium instability.

5 Discussion

Contrasting with control strategies like finite-time fuzzy synchronization that rely on Lyapunov stability for bounded errors, the NewtonFormer leverages Hessian-vector products to utilize second-order curvature, enabling faster convergence and escape from local minima in high-dimensional functions compared to first-order methods. Regarding scalability and robustness, unlike traditional controllers prone to parameter explosion, the model's multi-head attention processes high-dimensional features in parallel. In noisy or partially observed conditions, this mechanism acts as a "soft filter" to focus on key features, maintaining data-driven robustness without strict physical model constraints.

The proposed model shows potential for complex systems like autonomous navigation and robotic control. Specifically, the NTF model is suitable for domains requiring high-precision nonlinear prediction. In high-frequency financial trading, its fast convergence adapts to rapid market changes. For industrial predictive maintenance, it efficiently processes sensor time-series data for early fault warning. The model also aids scientific computing, improving forecast reliability by integrating physical constraints. Its robustness regarding data efficiency and noise offers distinct advantages in scenarios with data acquisition challenges. Unlike model-dependent strategies such as flatness-based control, our data-driven approach uses multi-head attention to capture long-term dynamics and Newton-CG for rapid adaptation in high dimensions. Regarding sensitivity, Newton-CG leverages curvature information to optimize step directions, significantly reducing dependence on learning rate fine-tuning compared to first-order methods. Furthermore, the architecture's parallel feature extraction ensures consistent convergence across random initializations, demonstrating robust training stability.

An in-depth comparison with baseline models reveals the root cause of NTF's superior performance. Compared to BP networks and LSTM, NTF's Transformer-based self-attention mechanism fundamentally addresses long-term dependency modeling and gradient issues, enabling direct capture of global nonlinear relationships. Compared to the standard Transformer, the periodically introduced Newton-CG steps leverage loss function curvature information to

provide a superior descent direction, leading to significantly accelerated convergence and escape from local optima. Compared to the pure Newton-CG optimization method, NTF, through data-driven feature learning, adaptively improves the conditioning of the optimization problem itself, thereby gaining strong generalization power rather than merely solving a single instance. Thus, NTF's success stems from the deep integration and synergy between representation learning and the optimization process.

Limitations and Future Work: The proposed hybrid architecture involves a trade-off between theoretical assumptions and computational efficiency, and its effectiveness in more complex, non-numerical modalities remains to be verified. Future efforts will focus on developing adaptive optimization mechanisms to reduce hyperparameter reliance and extending the framework to broader domains such as cross-modal learning and dynamic systems.

6 Conclusion

The nonlinear function-fitting prediction model based on the Newton-CG method and the Transformer proposed in this study demonstrates significant advantages in this task. Experimental data show that the mean square error of the NTF model on the test set is as low as 0.023, which is 71.5% and 68.2% higher than that of the traditional LSTM and BP networks respectively; The convergence speed improves by 60%, the number of iterations is reduced by 40%, and a good balance between efficiency and accuracy is achieved.

(1) The NTF model performs well on the core evaluation indicators, with an MSE of 0.023, an MSE of 0.015 for the Transformer model, and an MSE of 0.025 for the Newton-CG method, all of which are significantly better than the linear regression and neural network models. In terms of average absolute error, the MAE of the Transformer model is 0.08, and the MAE of the Newton-CG method is 0.12, while the residual distribution of the NTF model is highly concentrated at the value of 0, the dispersion is 65% lower than that of the traditional regression, and the R^2 score is as high as 0.99, indicating that the fitting ability of the model to nonlinear functions is close to perfect.

(2) The introduction of the Newton-CG algorithm reduces the number of training iterations by about 40% and the convergence time by 60% on average. Experiments show that the loss value of the NTF model drops below 10^{-4} and tends to be stable after 4 iterations, while the loss value of the standard Transformer model still fluctuates around 0.1 after 16 iterations. In terms of computational efficiency, the single-iteration time of NTF is about 1-2 seconds, and convergence typically requires only 30-50 iterations. When the memory is used about 3-4GB, the final loss value is stable in the range of 0.020-0.025. The efficiency is better than the conjugate gradient method and Hessian-Free method, and better than the pure Newton-CG method in terms of memory efficiency and computational accuracy.

(3) In terms of prediction error dimension, the error range of the NTF model is strictly controlled in the ±35 interval, which improves the prediction accuracy by 72% and 43% compared with Newton-CG and Transformer, respectively. Compared with the LSTM and MLP models, the NTF model shows better convergence stability and long-term prediction performance. The comparison of optimiser stability shows that Newton-CG has optimal stability, with an R^2 value close to 0.95 and the smallest fluctuation range, which is significantly better than traditional optimisers such as SGD and Adagrad.

To sum up, the NTF model constructed in this study achieves a triple breakthrough in high accuracy, strong generalization, and high efficiency in the task of nonlinear function fitting and prediction, and provides a new theoretical method and tool for modelling and predicting complex nonlinear systems. While the NTF model demonstrates excellent performance in nonlinear fitting tasks, this study has limitations that point to directions for future exploration. The current theoretical analysis relies on standard smoothness assumptions; future work could aim to analyze its convergence under more relaxed conditions. The Newton-CG step introduces additional computational overhead, and investigating its efficient integration with adaptive network architectures is an important direction. Extending the NTF framework to more challenging scenarios, such as online learning, federated learning, or solving partial differential equations involving physical conservation laws, would significantly broaden its application scope. Exploring a more fundamental theoretical connection between the attention mechanism and second-order optimization information may lead to more efficient novel hybrid architectures.

References

[1] Alqahtani, M. "Nonlinear autoregressive prediction model for VAWT power supply network energy management,". Energy Reports, vol.13, pp.5446-5462, 2025.

[2] Atila, H., & Spence, S. M. J. "Metamodeling of the response trajectories of nonlinear stochastic dynamic systems using physics-informed LSTM networks,". Journal of Building Engineering, vol.111, pp.113447, 2025.

[3] Bai, H., Zhang, J., Sun, K., & Kang, W.-H. "Nonlinear long-term vibration response prediction of offshore wind turbines under full operating conditions based on deep learning,". Applied Ocean Research, vol.159, pp.104625, 2025.

[4] Bu, Z., Long, B., Liu, Z., Wu, K., Geng, H., & Cheng, Y. "Multivariate adaptive Brownian Motion-Particle Filter framework for remaining useful life prediction of nonlinear and state-noise coupled degradation process,". Reliability Engineering & System Safety, vol.264, pp.111356, 2025.

[5] Cheng, Y., Fang, G., Cui, W., Li, Y., & Zhao, L. "Nonlinear flutter critical state prediction for a bridge girder based on instantaneous power balance principle,". Engineering Structures, vol.326, pp.119526, 2025.

[6] Derouech, Y., & Mesbahi, A. "Optimized estimation of Li-ion battery parameters to improve the Enhanced Self-Correcting model with nonlinear least-squares data fitting and SoC estimation using Invariant EKF,". e-Prime - Advances in Electrical Engineering, Electronics and Energy, vol.14, pp.101115, 2025.

[7] Ding, S., Shen, X., & Cai, Z. "Data-driven multi-step solar photovoltaic predictions with limited and uncertain information: Insights from a collaboratively-optimized nonlinear grey Bernoulli model,". Expert Systems with Applications, vol.258, pp.125170, 2024.

[8] Fan, M.-H., Zhao, J.-H., Ding, L., Ma, X.-Y., & Fu, R.-L. "Predicting nonlinear dynamic systems by causal physics-informed neural networks with ResNet blocks,". Neurocomputing, vol.656, pp.131589, 2025.

[9] Gou, X., Mi, C., Yang, Y., & Zeng, B. "A nonlinear mixed-frequency grey prediction model with two-stage lag parameter optimization and its application,". Applied Mathematical modeling, vol.150, pp.116360, 2026.

[10] Hlophe, T., Adcock, T. A. A., Ding, H., Zang, J., Dai, S., Tang, T., & Taylor, P. H. "Nonlinear wave loads on monopile foundations and structural response in severe wave conditions,". Applied Ocean Research, vol.164, pp.104790, 2025.

[11] Huangfu, N., Zhang, Y., Lei, Y., Liu, Q., Liao, W.-H., Zhao, Z., & Cao, J. "A dynamic response prediction model of nonlinear hysteretic isolators based on quasi-static characteristics and dynamic compensations,". Mechanical Systems and Signal Processing, vol.224, pp.112170, 2025.

[12] Jia, Q., Kang, K., Wang, B., Wu, Y., Zhang, Y., & Guo, F. "Research on wind power prediction with secondary decomposition and multi-algorithm fusion for complex nonlinear time series,". Computers and Electrical Engineering, vol.128, pp.110688, 2025.

[13] Jiang, P., Ren, W., Chen, Z., Wang, Z., Li, Y., & Dong, L. "A nonlinear dynamic ensemble remaining useful life prediction method considering multi-source data uncertainty,". Mechanical Systems and Signal Processing, vol.230, pp.112607, 2025.

[14] Khtir, S. M. A., Zahaf, H., Alla, H., & Velarde, M. G. "Colloidal particles: Further evidence of validity and usefulness of energies and interaction forces obtained from solving the nonlinear Poisson-Boltzmann equation,". Chemical Engineering Science, vol.320, pp.122473, 2026.

[15] Krack, M., Brake, M. R. W., Schwingshackl, C., Gross, J., Hippold, P., Lasen, M., Dini, D., Salles, L., Allen, M. S., Shetty, D., Payne, C. A., Willner, K., Lengger, M., Khan, M. Y., Ortiz, J., Najera-Flores, D. A., Kuether, R. J., Miles, P. R., Xu, C., … Scheel, M. "The Tribomechadynamics Research Challenge: Confronting blind predictions for the linear and nonlinear dynamics of a thin-walled jointed structure with measurement results,". Mechanical Systems and Signal Processing, vol.224, pp.112016, 2025.

[16] Leroux, C.-E., Fontvieille, C., & Bardin, F. "Including the nonlinear response of neurons to improve the prediction of visual acuity across levels of contrast, luminance, and blur,". Vision Research, vol.234, pp.108652, 2025.

[17] Li, X., Mao, Y., Li, C., & Mba, D. "Designing of a nonlinearly fused health indicator for incipient fault detection and degradation modeling using quadratic programming,". Applied Acoustics, vol.231, pp.110461, 2025.

[18] Liu, K., Gu, J., He, X., & Jia, J. "Safety-critical motion optimization for quadruped robots on offshore platforms: A hierarchical nonlinear model predictive control framework based on foothold optimization and control barrier function,". Control Engineering Practice, vol.165, pp.106559, 2025.

[19] Matsui, S., & Oka, M. "Long-term prediction of nonlinear ship roll motion using RAO,". Applied Ocean Research, vol.158, pp.104590, 2025.

[20] Mu, T., Li, R., Linghu, C., Liu, Y., Leng, J., Gao, H., & Hsia, K. J. "Nonlinear contact mechanics of soft elastic spheres under extreme compression,". Journal of the Mechanics and Physics of Solids, vol.203, pp.106229, 2025.

[21] Orito, Y. "Accurate and Scalable Prediction of a Fast and Highly Exothermic Nonlinear Reaction System: Reaction Development Using Coupled Simulation of a Mechanism-Oriented Kinetic Model and a Customized Heat Removal Model,". Organic Process Research & Development, vol.29, no.7, pp.1757-1765, 2025.

[22] Oskui, A. E., Cao, J., Sadeghzade, S., & Yuan, H. "Influence of loading mode on the prediction accuracy of nonlinear viscoelastic models for soft adhesives,". International Journal of Solids and Structures, vol.322, pp.113595, 2025.

[23] Qin, F., Tong, M., Huang, Y., & Zhang, Y. "Modeling, prediction and analysis of natural gas consumption in China using a novel dynamic nonlinear multivariable grey delay model,". Energy, vol.305, pp.132105, 2024.

[24] Qu, M., Jasim, D. J., Alizadeh, A., Eftekhari, S. A., Nasajpour-Esfahani, N., Zekri, H., Salahshour, S., & Toghraie, D. "A new model for viscosity prediction for silica-alumina-MWCNT/Water hybrid nanofluid using nonlinear curve fitting,". Engineering Science and Technology, an International Journal, vol.50, pp.101604, 2024.

[25] Rong, P., Zuo, Y., Lin, J., Zheng, L., Pan, C., Sun, W., Chen, Q., & Chen, B. "In situ stress prediction model in complex geology: A hybrid GA-ANN with nonlinear boundary condition,". Journal of Rock Mechanics and Geotechnical Engineering, vol.17, no.7, pp.4349-4366, 2025.

[26] Shang, Y., Qu, D., Li, J., & Zhang, R. "A new parameter-free continuously differentiable filled function algorithm for solving nonlinear equations and data fitting problems,". Journal of Computational and Applied Mathematics, vol.454, pp.116198, 2025.

[27] Shi, W., Wan, X., Zhao, F., & Deng, R. "A dual-model framework combining nonlinear autoregressive with exogenous inputs (NARX) and LSTM networks for enhanced daily runoff

prediction and error correction,". Environmental modeling & Software, vol.192, pp.106570, 2025.
[28] Sun, Z., Xie, T., Li, M., & Guo, T. "Quasi-real-time prediction and warning of stay cable vibration subjected to typhoon with a Nonlinear Autoregressive Neural Network and KNN,". Engineering Structures, vol.340, pp.120661, 2025.
[29] Tian, W., Wang, W., Wang, Y., Shi, C., & Ma, Q. "Accurate runoff prediction in nonlinear and nonstationary environments using a novel hybrid model,". Journal of Hydrology, vol.662, pp.133949, 2025.
[30] Wang, P., Wang, J., Li, Z., Qu, T., Xu, F., Hu, Y., & Yang, H. "Koopman-MPC-based energy-efficient integrated control of attitude maneuver and vibration suppression for nonlinear in-wheel motor-active suspension on uneven roads,". Energy, vol.336, pp.138157, 2025.
[31] Almazroi, A. A., Alkinani, M. H., Al-Shareeda, M. A., & Manickam, S. "A Novel DDoS Mitigation Strategy in 5G-Based Vehicular Networks Using Chebyshev Polynomials,". ARABIAN JOURNAL FOR SCIENCE AND ENGINEERING, vol.49no.9, pp.11991 ~ 12004, 2024.
[32] Almazroi, A. A., Alqarni, M. A., Al-Shareeda, M. A., Alkinani, M. H., Almazroey, A. A., & Gaber, T. "FCA-VBN: Fog computing-based authentication scheme for 5G-assisted vehicular blockchain network,". INTERNET OF THINGS, vol.25, 2024.