Uluslararası İleri Doğa Bilimleri ve Mühendislik Araştırmaları Dergisi
Sayı 8, S. 846-858, 11, 2024

Araştırma Makalesi

IJANSER
https://as-proceeding.com/index.php/ijanser
ISSN:2980-0811

International Journal of Advanced Natural Sciences and Engineering Researches
Volume 8, pp. 846-858, 11, 2024

Research Article

Modeling and Optimization of Copper Tube Drawing Process Parameters Based on Neuro-Regression and Stochastic Search Techniques

Melih Savran*

[1]*Department of Mechanical Engineering/İzmir Katip Çelebi University, Turkey*

**mlhsvrn@gmail.com*

(Received: 11 December 2024, Accepted: 29 December 2024)

(5th International Conference on Scientific and Academic Research ICSAR 2024, December 23-24, 2024)

ATIF/REFERENCE: Savran, M. (2024). Modeling and Optimization of Copper Tube Drawing Process Parameters Based on Neuro-Regression and Stochastic Search Techniques. *International Journal of Advanced Natural Sciences and Engineering Researches*, 8(11), 846-858.

Abstract **–**This study is dedicated to optimizing two critical responses within the tube drawing process: drawing force and linear thickness distribution. The process is influenced by four design factors: die angle, bearing length, friction coefficient, and drawing velocity.

To mathematically define the tube drawing process, 13 functional forms were utilized, encompassing linear, quadratic, trigonometric, and logarithmic expressions and their rational and hybrid combinations. The dataset for model development was taken from the literature. The candidate models were assessed using multiple performance metrics, including R^2 training, R^2 testing, and R^2 validation, along with boundedness checks and adherence to predefined constraints.

After identifying a suitable model, modified version of stochastic search optimization methods; Differential Evolution and Simulated Annealing were applied to minimize drawing force while maximizing thickness distribution. The results revealed a minimum drawing force of 2107.75 kN and a maximum linear thickness distribution of 0.911 mm.

These findings underscore the robustness and versatility of the neuro-regression modeling and stochastic optimization processes proposed in this study. In comparison to previous methodologies—such as response surface and artificial bee optimization techniques referenced in the literature—this approach has yielded an improvement of 3.5 % in drawing force and an enhancement of 18% in thickness distribution.

Keywords – Tube drawing process, neuro-regression modeling, stochastic optimization, copper tube

I. INTRODUCTION

The tube drawing process is a widely utilized metal forming technique for producing tubes with precise dimensions, improved surface quality, and enhanced mechanical properties. Due to the complex interactions between process parameters, developing accurate mathematical models to predict critical responses such as drawing force and thickness distribution is essential for optimizing the process and ensuring product quality [1]. Mathematical modeling serves as a fundamental tool in understanding these relationships and forms the basis for optimization techniques aimed at improving process efficiency and outcomes.

Recent advancements in computational methods have facilitated the development of sophisticated models that incorporate linear, nonlinear, and hybrid mathematical functions. These models are invaluable for capturing the nonlinearities inherent in the tube drawing process, as highlighted in studies such as those by Pater et al. [2] and Tiesler et al. [3]. Factors such as die angle, bearing length, friction coefficient, and drawing velocity significantly influence process responses, necessitating the use of advanced modeling techniques to accurately represent their effects [4].

Response Surface Methodology (RSM), Artificial Neural Networks (ANNs), Taguchi, Central Composite Design (CCD), and Genetic Algorithms (GAs) have been extensively applied in this domain to optimize process parameters. For instance, Kumar and Chauhan [5] employed RSM to optimize drawing velocity, die angle, and friction coefficient, achieving a reduction in drawing force by 15%. Similarly, ANNs have been utilized to model the nonlinear behavior of process parameters with high accuracy, as demonstrated by Rajput and Nimbalkar [6], where the model predicted thickness distribution with an error margin below 2%.

The Taguchi method, often integrated with ANOVA, has proven effective in identifying significant factors influencing process responses. Patel et al. [7] used the Taguchi approach to optimize die design and process velocity, resulting in a 12% improvement in thickness uniformity. CCD, a design of experiments approach, has also been widely used to explore the interactive effects of process variables. Yadav et al. [8] utilized CCD to develop second-order regression models for drawing force and achieved a significant reduction in force requirements.

In addition, heuristic algorithms have been employed to identify global optima for complex, multi-objective problems in the tube drawing process. Sun et al. [9] demonstrated the application of GAs for optimizing die geometry and bearing length, achieving a 10% increase in production efficiency while maintaining product quality.

This study aims to develop mathematical models for the copper tube drawing process using a diverse set of functional forms, including linear, quadratic, trigonometric, and logarithmic expressions, as well as their rational and hybrid variants. The models are evaluated based on performance metrics such as R^2training, R^2 testing, and R^2 validation along with boundedness checks and compliance with predefined constraints. Following model selection, optimization techniques are employed to minimize drawing force and maximize thickness distribution. By bridging neuro regression mathematical modeling and stochastic optimization, this study seeks to contribute to the field of process design and control in metal forming operations.

II. Materials and Method

Neuro-regression is a sophisticated mathematical modeling technique that combines artificial neural networks with regression analysis. The dataset is divided into training, testing, and validation subsets to facilitate this modeling process. In the training phase, 80% of the dataset is used to determine the model coefficients that yield the highest accuracy in predicting actual values.

The remaining 20% of the dataset is reserved for testing and validation, allowing for an evaluation of the model's predictive performance on unseen data. Tables 1 and 2 present the dataset and proposed mathematical models used to analyze the relationship between input and output parameters during the mathematical modeling phase using the neuro-regression approach.

Table 1. The data set formed using the Box–Behnken experimental design method [10]

Run Order	x1 (Deg)	x2 (mm)	x3 (-)	x4 (mm/min)	Drawing Force (kN)	Linear Thickness (mm)
1	5	4	0.2	7.5	4752	0.72757
2	10	4	0.2	7.5	3379	0.73827
3	5	8	0.2	7.5	4763	0.72955
4	10	8	0.2	7.5	3477	0.73741
5	7.5	6	0.1	5	2358	0.73511
6	7.5	6	0.3	5	5277	0.73744
7	7.5	6	0.1	10	2371	0.73515
8	7.5	6	0.3	10	5284	0.7358
9	5	6	0.2	5	4748	0.72218
10	10	6	0.2	5	3415	0.74366
11	5	6	0.2	10	4760	0.72669
12	10	6	0.2	10	3433	0.74371
13	7.5	4	0.1	7.5	2314	0.72837
14	7.5	8	0.1	7.5	2386	0.72181
15	7.5	4	0.3	7.5	5183	0.72727
16	7.5	8	0.3	7.5	5442	0.72164
17	5	6	0.1	7.5	2743	0.72175
18	10	6	0.1	7.5	2180	0.73938
19	5	6	0.3	7.5	6864	0.72932
20	10	6	0.3	7.5	4675	0.74629
21	7.5	4	0.2	5	3744	0.7274
22	7.5	8	0.2	5	3927	0.72202
23	7.5	4	0.2	10	3724	0.72781
24	7.5	8	0.2	10	3951	0.7218
25	7.5	6	0.2	7.5	3826	0.73529

The effectiveness of the model's predictions is assessed using the R^2 criterion, where a score approaching 1 indicates outstanding performance. Furthermore, boundedness check criteria are applied to ensure that the model's outputs are realistic and viable from an engineering standpoint.

The optimal model is expected to achieve two fundamental objectives: (1) attain high predictive accuracy, with an R^2 value as close to 1 as possible, and (2) produce designs that are realistic and practical within an engineering context. The model that meets these criteria is chosen as the objective function for optimization, and its value is either maximized or minimized depending on the specific optimization goal.

Table 2. Mathematical model types including linear, quadratic, trigonometric, logarithmic, rational and their hybrid forms [11].

Model Name	Nomenclature	Formula
Multiple linear	L	$Y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \alpha_4 x_4$
Multiple linear rational	LR	$Y = (\alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \alpha_4 x_4)/(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4)$
Second order multiple nonlinear	SON	$Y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \alpha_4 x_4 + \alpha_5 x_1^2 + \alpha_6 x_2^2 + \alpha_7 x_3^2 + \alpha_8 x_4^2 + \alpha_9 x_1 x_2 + \alpha_{10} x_1 x_3 + \alpha_{11} x_1 x_4 + \alpha_{12} x_2 x_3 + \alpha_{13} x_2 x_4 + \alpha_{14} x_3 x_4$
Second order multiple nonlinear rational	SONR	$Y = (\alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \alpha_4 x_4 + \alpha_5 x_1^2 + \alpha_6 x_2^2 + \alpha_7 x_3^2 + \alpha_8 x_4^2 + \alpha_9 x_1 x_2 + \alpha_{10} x_1 x_3 + \alpha_{11} x_1 x_4 + \alpha_{12} x_2 x_3 + \alpha_{13} x_2 x_4 + \alpha_{14} x_3 x_4)/(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_1^2 + \beta_6 x_2^2 + \beta_7 x_3^2 + \beta_8 x_4^2 + \beta_9 x_1 x_2 + \beta_{10} x_1 x_3 + \beta_{11} x_1 x_4 + \beta_{12} x_2 x_3 + \beta_{13} x_2 x_4 + \beta_{14} x_3 x_4)$
First order trigonometric multiple nonlinear	FOTN	$Y = \alpha_0 + \alpha_1 \sin x_1 + \alpha_2 \sin x_2 + \alpha_3 \sin x_3 + \alpha_4 \sin x_4 + \alpha_5 \cos x_1 + \alpha_6 \cos x_2 + \alpha_7 \cos x_3 + \alpha_8 \cos x_4$
First order trigonometric multiple nonlinear rational	FOTNR	$Y = (\alpha_0 + \alpha_1 \sin x_1 + \alpha_2 \sin x_2 + \alpha_3 \sin x_3 + \alpha_4 \sin x_4 + \alpha_5 \cos x_1 + \alpha_6 \cos x_2 + \alpha_7 \cos x_3 + \alpha_8 \cos x_4)/(\beta_0 + \beta_1 \sin x_1 + \beta_2 \sin x_2 + \beta_3 \sin x_3 + \beta_4 \sin x_4 + \beta_5 \cos x_1 + \beta_6 \cos x_2 + \beta_7 \cos x_3 + \beta_8 \cos x_4)$
Second order trigonometric multiple nonlinear	SOTN	$Y = \alpha_0 + \alpha_1 \sin x_1 + \alpha_2 \sin x_2 + \alpha_3 \sin x_3 + \alpha_4 \sin x_4 + \alpha_5 \cos x_1 + \alpha_6 \cos x_2 + \alpha_7 \cos x_3 + \alpha_8 \cos x_4 + \alpha_9 \sin^2 x_1 + \alpha_{10} \sin^2 x_2 + \alpha_{11} \sin^2 x_3 + \alpha_{12} \sin^2 x_4 + \alpha_{13} \cos^2 x_1 + \alpha_{14} \cos^2 x_2 + \alpha_{15} \cos^2 x_3 + \alpha_{16} \cos^2 x_4$
Second order trigonometric multiple nonlinear	SOTNR	$Y = (\alpha_0 + \alpha_1 \sin x_1 + \alpha_2 \sin x_2 + \alpha_3 \sin x_3 + \alpha_4 \sin x_4 + \alpha_5 \cos x_1 + \alpha_6 \cos x_2 + \alpha_7 \cos x_3 + \alpha_8 \cos x_4 + \alpha_9 \sin^2 x_1 + \alpha_{10} \sin^2 x_2 + \alpha_{11} \sin^2 x_3 + \alpha_{12} \sin^2 x_4 + \alpha_{13} \cos^2 x_1 + \alpha_{14} \cos^2 x_2 + \alpha_{15} \cos^2 x_3 + \alpha_{16} \cos^2 x_4)/(\beta_0 + \beta_1 \sin x_1 + \beta_2 \sin x_2 + \beta_3 \sin x_3 + \beta_4 \sin x_4 + \beta_5 \cos x_1 + \beta_6 \cos x_2 + \beta_7 \cos x_3 + \beta_8 \cos x_4 + \beta_9 \sin^2 x_1 + \beta_{10} \sin^2 x_2 + \beta_{11} \sin^2 x_3 + \beta \sin^2 x_4 + \beta_{13} \cos^2 x_1 + \beta_{14} \cos^2 x_2 + \beta_{15} \cos^2 x_3 + \beta_{16} \cos^2 x_4)$
First order logarithmic multiple nonlinear	FOLN	$Y = \alpha_0 + \alpha_1 lnx_1 + \alpha_2 lnx_2 + \alpha_3 lnx_3 + \alpha_4 lnx_4$
First order logarithmic multiple nonlinear rational	FOLNR	$Y = (\alpha_0 + \alpha_1 lnx_1 + \alpha_2 lnx_2 + \alpha_3 lnx_3 + \alpha_4 lnx_4)/(\beta_0 + \beta_1 lnx_1 + \beta_2 lnx_2 + \beta_3 lnx_3 + \beta_4 lnx_4)$
Second order logarithmic multiple nonlinear	SOLN	$Y = \alpha_0 + \alpha_1 lnx_1 + \alpha_2 lnx_2 + \alpha_3 lnx_3 + \alpha_4 lnx_4 + \alpha_5 ln^2 x_1 + \alpha_6 ln^2 x_2 + \alpha_7 ln^2 x_3 + \alpha_8 ln^2 x_4 + \alpha_9 lnx_1 x_2 + \alpha_{10} lnx_1 x_3 + \alpha_{11} lnx_1 x_4 + \alpha_{12} lnx_2 x_3 + \alpha_{13} lnx_2 x_4 + \alpha_{14} lnx_3 x_4$
Second order logarithmic multiple nonlinear rational	SOLNR	$Y = (\alpha_0 + \alpha_1 lnx_1 + \alpha_2 lnx_2 + \alpha_3 lnx_3 + \alpha_4 lnx_4 + \alpha_5 ln^2 x_1 + \alpha_6 ln^2 x_2 + \alpha_7 ln^2 x_3 + \alpha_8 ln^2 x_4 + \alpha_9 lnx_1 x_2 + \alpha_{10} lnx_1 x_3 + \alpha_{11} lnx_1 x_4 + \alpha_{12} lnx_2 x_3 + \alpha_{13} lnx_2 x_4 + \alpha_{14} lnx_3 x_4)/(\beta_0 + \beta_1 lnx_1 + \beta_2 lnx_2 + \beta_3 lnx_3 + \beta_4 lnx_4 + \beta_5 ln^2 x_1 + \beta_6 ln^2 x_2 + \beta_7 ln^2 x_3 + \beta_8 ln^2 x_4 + \beta_9 lnx_1 x_2 + \beta_{10} lnx_1 x_3 + \beta_{11} lnx_1 x_4 + \beta_{12} lnx_2 x_3 + \beta_{13} lnx_2 x_4 + \beta_{14} lnx_3 x_4)$
Hybrid model	H	$Y = a_0 + \alpha_1 Sinx_1 + a_2 Sinx_1^2 + a_3 Sinx_2 + a_4 Sinx_2^2 + \alpha_5 Sinx_3 + a_6 Sinx_3^2 + a_7 Sinx_4 + a_8 Sinx_4^2 + a_9 Sin[x_1] Cos[x_2] + a_{10} Sin[x_1] Cos[x_3] + a_{11} Sin[x_1] Cos[x_4] + a_{12} Sin[x_2] Cos[x_3] + a_{13} Sin[x_2] Cos[x_4] + a_{14} Sin[x_3] Cos[x_4] + a_{15} x_1 + a_{16} x_1^2 + a_{17} x_2 + a_{18} x_2^2 + a_{19} x_3 + a_{20} x_3^2 + a_{21} x_4 + a_{22} x_4^2$

In the optimization process, the output parameters—drawing force and thickness—will be minimized and maximized, respectively. To achieve this, modified versions of the differential evolution and simulated annealing algorithms, available within the Mathematica software, will be employed.

The Differential Evolution (DE) algorithm in Mathematica is an evolutionary optimization method designed for solving complex, nonlinear, and multidimensional optimization problems. It operates by generating an initial population of candidate solutions and iteratively refining them through mutation, crossover, and selection processes to minimize or maximize the target objective function. Mutation creates new solutions by combining existing ones using a weighted difference. Crossover blends the mutated solution with a current population member to introduce variability. Selection compares the new candidate with its counterpart in the population and retains the better solution. These processes drive the algorithm toward optimal solutions [12].

Simulated Annealing (SA) in Mathematica is a probabilistic optimization algorithm inspired by the annealing process in metallurgy. The algorithm explores the solution space by probabilistically accepting worse solutions to escape local optima, gradually reducing the probability as the process progresses. The algorithm has several adjustable options, enhancing its applicability. "BoltzmannExponent" controls temperature decay, and "InitialPoints" defines starting locations. "LevelIterations" sets the number of iterations per temperature level, while "PerturbationScale" adjusts step sizes for exploration. "RandomSeed" ensures reproducibility, and "SearchPoints" determines the number of points evaluated, balancing thoroughness and computational cost. These options enable tailored optimization for diverse applications [13].

The objective function, constraints, design variables, and related details for the single-objective optimization problem addressed in this study are as follows.

Single-Objective Problem

Find
$\{\text{Die Angle } (x1) \in [5, 10] \wedge \text{Bearing Length } (x2) \in [4, 8] \wedge \text{Friction Coefficient } (x3) \in [0.1,\ 0.3] \wedge$
$\text{Drawing Velocity } (x4) \in [5,\ 10]$

Minimize
Drawing Force $(x1, x2, x3, x4)$

Maximize
Linear Thickness $(x1, x2, x3, x4)$

Constraints

Scenario 1

$5 \leq \{x1\} \leq 10, 4 \leq \{x2\} \leq 8, 0.1 \leq \{x3\} \leq 0.3, 5 \leq \{x4\} \leq 10$

Scenario 2

$5 \leq \{x1\} \leq 10, 4 \leq \{x2\} \leq 8, 0.1 \leq \{x3\} \leq 0.3, 5 \leq \{x4\} \leq 10, \{x1, x2, x4\} \in \text{Integers}$

Scenario 3
$\{x1\} \in \{5, 7.5, 10\}, \{x2\} \in \{4, 6, 8\}, \{x3\} \in \{0.1, 0.2, 0.3\}, \{x4\} \in \{5, 7.5, 10\}$

Design variables
x1, x2, x3, x4

Mathematical modeling and optimization steps regarding tube drawing process are given in Figure 1. During the solution phase of the problem discussed in the study, the steps in the flow chart were followed to determine the best designs for different scenarios.

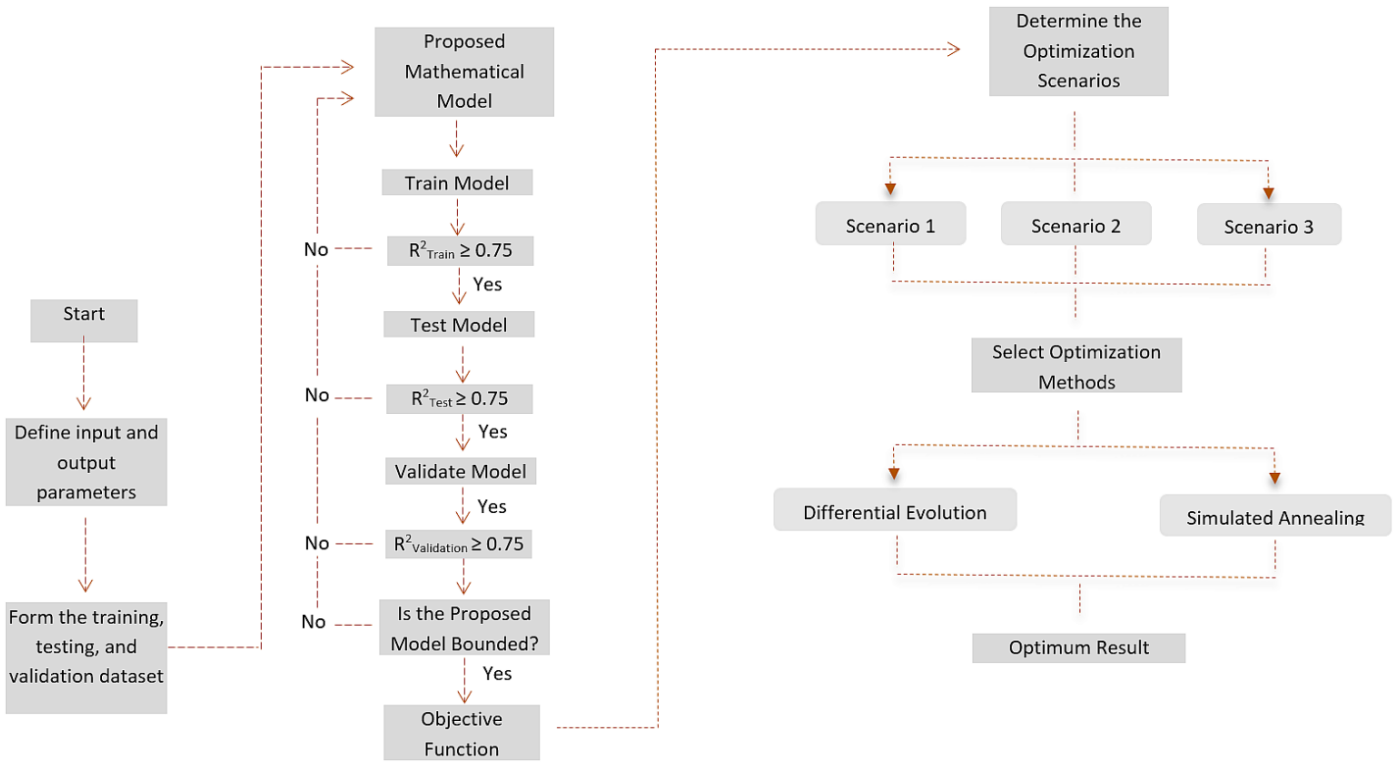

Figure 1. Flowchart of mathematical modelling and optimization process

III. Results and Discussion

Mathematical modeling and optimization were performed using neuro-regression and stochastic search methods, respectively. The prediction performance evaluation of the proposed models is given in Table 3 (drawing force) and Table 4 (linear thickness) to determine which one best describes the relationship between the design and output parameters.

The performance of the neuro-regression models for predicting drawing force varies significantly across metrics. The R^2 values for training indicate high accuracy for most models, such as LR (0.999821) and SON (0.999939), demonstrating effective training phases. However, models like SONR and SOTNR exhibit extremely poor testing and validation performance, with negative R^2 values (e.g., -26.1326 and -29.6735, respectively), suggesting overfitting or inconsistent predictions. Notably, these models also produce extreme max and min values, such as 3.29289×10^{9} and -3.28×10^{15}, indicating numerical instability.

Conversely, FOLNR and LR achieve robust results with R^2 values near 1 for both testing (0.99) and validation (0.99), along with stable max and min values. This indicates their suitability for predictive modeling in this context. FOLNR was selected as the objective function for the optimization phase because its minimum value indicates a significant potential for reducing drawing force, ultimately enhancing our design efficiency.

Table 3. Results of the Neuro-regression models for the drawing force

Model*	R^2Training	R^2Testing	R^2Validation	Max	Min
L	0.997	0.720	0.979	6344.640	1452.040
LR	0.999	0.988	0.985	7133.920	2261.850
SON	0.999	0.604	0.965	7140.880	1990.210
SONR	0.657	-26.132	-11.314	3.292*10^9	-3.280*10^15
FOTN	0.998	0.784	0.962	6949.350	822.995
FOTNR	0.217	-29.428	-11.724	2.501*10^7	-253126
SOTN	0.998	0.784	0.962	6551.390	728.951
SOTNR	0.288	-29.673	-11.731	2.209*10^10	-1.180*10^15
FOLN	0.996	0.669	0.948	6292.820	1278.930
FOLNR	0.999	0.998	0.989	7040.160	2107.750
SOLN	0.998	0.784	0.962	6495.560	1427.470
SOLNR	0.765	-5.863	-3.505	4.451*10^6	-839122
H	0.999	0.163	0.952	7909.840	397.363

*The full form of the models are given in Table 7

For linear thickness predictions, the models again exhibit a wide performance range (Table 4). While models like FOTN, SOTN, and SOLN maintain high R^2 values across training, testing, and validation phases (e.g., R^2 for testing in FOTN = 0.783396), others like H and FOLNR struggle with extreme negative testing R^2 values (-29.8284 and -7.67847, respectively). This reflects their inability to generalize effectively.

Notably, some models, such as SOTNR, produce extreme output ranges (e.g., max = 1.64689×10^6, min = −82558.5), further emphasizing numerical or parameter instability. Stable models like FOTNR and SOLNR achieve consistent high R^2 values across metrics, with moderate max and min values, confirming their reliability.

SOTN has the same prediction performance as FOTN, SOLN models and performs better regarding maximum linear thickness. However, to examine whether the SOTN, FOTN, and SOLN models will produce similar results in terms of minimum linear thickness under different optimization scenarios, all three were selected as separate objective functions.

Table 4. Results of the Neuro-regression models for the linear thickness

Model*	R^2Training	R^2Testing	R^2Validation	Max	Min
L	0.999	0.572	0.424	0.741	0.721
LR	0.999	0.126	-482.807	61589.700	-8.11*10^10
SON	0.999	0.316	0.826	0.752	0.712
SONR	0.999	0.261	0.821	0.7515	0.711
FOTN	0.999	0.783	0.926	0.748	0.712
FOTNR	0.999	0.744	0.867	0.749	0.691
SOTN	0.999	0.783	0.926	0.911	0.682
SOTNR	0.999	0.0109	0.521	1.646*10^6	-82558.5
FOLN	0.999	0.475	0.318	0.740	0.721
FOLNR	0.999	-7.678	0.475	100.465	-1.48*10^12
SOLN	0.999	0.783	0.926	0.744	0.718
SOLNR	0.999	0.787	0.915	0.745	0.719
H	0.999	-29.824	0.222	0.874	0.350

*The full form of the models are given in Table 8

The FOLNR model's optimization results reveal consistent performance across all scenarios (Table 5). Regardless of the constraints or optimization algorithm employed (Differential Evolution (DE) or

Simulated Annealing (SA)), the minimum drawing force achieved remains constant at 2107.15 kN. This outcome indicates an improvement of approximately 3.5% in drawing force compared to the results in the dataset. Furthermore, the recommended design from this model is same in the all optimization scenarios. This consistency highlights the FOLNR model's robustness in optimizing drawing force under varying conditions.

Table 5. Results of optimization problems for drawing force

Scenario Number	Constrains	Objective Function	Optimization Algorithm	Minimum Drawing Force	Suggested Design
1	$5 \leq x1 \leq 10$, $4 \leq x2 \leq 8$, $0.1 \leq x3 \leq 0.3$, $5 \leq x4 \leq 10$	FOLNR	DE	2107.15	$x_1: 10, x_2: 4, x_3: 0.1, x_4: 10,$
			SA	2107.15	$x_1: 10, x_2: 4, x_3: 0.1, x_4: 10,$
2	$5 \leq x1 \leq 10$, $4 \leq x2 \leq 8$, $0.1 \leq x3 \leq 0.3$, $5 \leq x4 \leq 10$, $\{x1, x2, x4\} \in$ Integers		DE	2107.15	$x_1: 10, x_2: 4, x_3: 0.1, x_4: 10,$
			SA	2107.15	$x_1: 10, x_2: 4, x_3: 0.1, x_4: 10,$
3	$x1 = 5 \,\|\| x1 = 7.5 \,\|\| x1 = 10$, $x2 = 4 \,\|\| x2 = 6 \,\|\| x2 = 8$, $x3 = 0.1 \,\|\| x3 = 0.2 \,\|\| x3 = 0.3$, $x4 = 5 \,\|\| x4 = 7.5 \,\|\| x4 = 10$		DE	2107.15	$x_1: 10, x_2: 4, x_3: 0.1, x_4: 10,$
			SA	2107.15	$x_1: 10, x_2: 4, x_3: 0.1, x_4: 10,$

Table 6 illustrates how the linear thickness varies based on design parameters, considering different optimization scenarios, mathematical models, and optimization methods. The optimization outcomes for three selected models (SOTN, FOTN, and SOLN) for linear thickness vary significantly depending on the scenario and algorithm used. For Scenario 1, SOTN achieves the highest maximum linear thickness of 0.911 mm with both DE and SA algorithms, outperforming FOTN and SOLN, which achieve maximum values of 0.748 and 0.745, respectively.

In Scenario 2, SOTN again performs well, with maximum thickness values of 0.899 (DE) and 0.875 (SA). In contrast, FOTN and SOLN maintain values similar to those in Scenario 1. By Scenario 3, all models converge to the same maximum linear thickness of 0.744, regardless of the optimization algorithm applied. This indicates a potential convergence or limitation in further optimizing thickness beyond this point.

Table 6. Results of optimization problems considering three selected models for linear thickness

Scenario Number	Constrains	Objective Function	Optimization Algorithm	Maximum Linear Thickness	Suggested Design
1	$5 \le x1 \le 10$, $4 \le x2 \le 8$, $0.1 \le x3 \le 0.3$, $5 \le x4 \le 10$	SOTN	DE	0.911	$x_1: 6.283, x_2: 4.934, x_3: 0.241, x_4: 6.276$
			SA	0.911	$x_1: 6.283, x_2: 4.934, x_3: 0.241, x_4: 6.276$
		FOTN	DE	0.748	$x_1: 9.179, x_2: 5.608, x_3: 0.242, x_4: 8.195$
			SA	0.748	$x_1: 9.179, x_2: 5.608, x_3: 0.242, x_4: 8.195$
		SOLN	DE	0.745	$x_1: 10, x_2: 5.420, x_3: 0.228, x_4: 7.426$
			SA	0.745	$x_1: 10, x_2: 5.420, x_3: 0.228, x_4: 7.426$
2	$5 \le x1 \le 10$, $4 \le x2 \le 8$, $0.1 \le x3 \le 0.3$, $5 \le x4 \le 10$, $\{x1, x2, x4\} \in$ Integers	SOTN	DE	0.899	$x_1: 6, x_2: 5, x_3: 0.2419, x_4: 6$
			SA	0.875	$x_1: 6, x_2: 6, x_3: 0.2419, x_4: 6$
		FOTN	DE	0.748	$x_1: 9, x_2: 6, x_3: 0.242, x_4: 8$
			SA	0.747	$x_1: 9, x_2: 6, x_3: 0.261, x_4: 9$
		SOLN	DE	0.744	$x_1: 10, x_2: 5, x_3: 0.228, x_4: 7$
			SA	0.744	$x_1: 10, x_2: 5, x_3: 0.255, x_4: 8$
3	$x1 = 5 \mid\mid x1 = 7.5 \mid\mid x1 = 10$, $x2 = 4 \mid\mid x2 = 6 \mid\mid x2 = 8$, $x3 = 0.1 \mid\mid x3 = 0.2 \mid\mid x3 = 0.3$, $x4 = 5 \mid\mid x4 = 7.5 \mid\mid x4 = 10$	SOTN	DE	0.744	$x_1: 10, x_2: 6, x_3: 0.2, x_4: 7.5$
			SA	0.744	$x_1: 10, x_2: 6, x_3: 0.2, x_4: 7.5$
		FOTN	DE	0.744	$x_1: 10, x_2: 6, x_3: 0.2, x_4: 7.5$
			SA	0.744	$x_1: 10, x_2: 6, x_3: 0.2, x_4: 7.5$
		SOLN	DE	0.744	$x_1: 10, x_2: 6, x_3: 0.2, x_4: 7.5$
			SA	0.744	$x_1: 10, x_2: 6, x_3: 0.2, x_4: 7.5$

IV. CONCLUSION

This study successfully demonstrated the application of neuro-regression modeling and modified stochastic optimization techniques to optimize two critical parameters in the tube drawing process: drawing force and thickness distribution. By systematically analyzing the influences of die angle, bearing length, friction coefficient, and drawing velocity, the research provided a comprehensive framework for improving process performance.

The models that best describe the input-output relationships for the tube drawing process were determined using 13 different mathematical functional forms. Performance metric R^2 and a boundedness check ensured reliability and consistency. The FOLNR and SOTN models minimized drawing force to 2107.75 kN and achieved a maximum thickness distribution of 0.911 mm.

Moreover, adopting Differential Evolution and Simulated Annealing algorithms proved instrumental in surpassing prior methodologies. Compared to existing techniques, such as response surface modeling and artificial bee optimization, they yielded a 3.5% improvement in drawing force and an 18% enhancement in thickness distribution.

These outcomes highlight the robustness and adaptability of the proposed neuro-regression and optimization framework, offering a valuable contribution to the tube drawing process. The methodology establishes a foundation for future design and optimization applications in metal forming and other manufacturing processes.

REFERENCES

[1] S. Kalpakjian and S. R. Schmid, Manufacturing Engineering and Technology. Pearson Education, 2014.
[2] Z. Pater, J. Kazanecki, and J. Bartnicki, Analysis of drawing force and energy efficiency in the tube drawing process, *International Journal of Mechanical Sciences*, vol. 122, pp. 304–315, 2017.
[3] N. Tiesler, P. Franke, and G. Hirt, Application of finite element modeling in the optimization of tube forming processes, *Materials Today Communications*, vol. 24, pp. 101–113, 2020.
[4] T. Altan and A. E. Tekkaya, Sheet Metal Forming: Processes and Applications. ASM International, 2012.
[5] R. Kumar and S. R. Chauhan, Optimization of tube drawing parameters using Response Surface Methodology, *Materials Today: Proceedings*, vol. 26, pp. 1527–1532, 2020.
[6] S. S. Rajput and P. Nimbalkar, Application of Artificial Neural Networks for modeling and optimizing tube drawing, *Journal of Intelligent Manufacturing*, vol. 30, no. 3, pp. 793–805, 2019.
[7] R. Patel, V. Sharma, and P. Kumar, Taguchi-based optimization of the tube drawing process with emphasis on die design, *Journal of Manufacturing Processes*, vol. 62, pp. 45–55, 2021.
[8] R. Yadav, R. K. Singh, and V. Mishra, Central Composite Design for optimization of tube drawing parameters, *International Journal of Advanced Manufacturing Technology*, vol. 97, no. 1, pp. 1023–1034, 2018.
[9] W. Sun, Y. Zhang, and H. Li, Optimization of tube drawing parameters using Genetic Algorithms for multi-objective solutions, *Journal of Materials Processing Technology*, vol. 249, pp. 290–298, 2017.
[10] M. Salehi, M. Hosseinzadeh, and M. Elyasi, A study on optimal design of process parameters in tube drawing process of rectangular parts by combining box–behnken design of experiment, response surface methodology and artificial bee colony algorithm, *Transactions of the Indian Institute of Metals*, vol. 69, pp. 1223–1235, 2016.
[11] İ. Polatoğlu, L. Aydın, B. Ç. Nevruz, and S. Özer, A novel approach for the optimal design of a biosensor, *Analytical Letters*, vol. 53, no. 9, pp. 1428–1445, 2020.
[12] M. Savran, L. Aydin, A. Ayaz, and T. Uslu, A new strategy for manufacturing, modeling, and optimization of 3D printed polylactide based on multiple nonlinear neuro regression analysis and stochastic optimization methods, *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, vol. 09544089241272909, 2024.
[13] M. Savran and L. Aydin, An Integrated Approach to Modeling and Optimization in Engineering and Science, CRC Press, 2024.

APPENDIX

Table 7. Full form mathematical models for drawing force (Related to Table 3)

L	2623.68 - 295.603 x1 + 54.5304 x2 + 15918.4 x3 - 2.55692 x4
LR	(-24469.9 + 23868.1 x1 - 5593.87 x2 + 1.61274*10^6 x3 + 445.591 x4)/(31.7107 + 13.3861 x1 - 2.27088 x2 - 22.8413 x3 + 0.307225 x4)
SON	-1343.06 - 551.637 x1 + 50.965 x1^2 + 750.359 x2 - 13.1625 x1 x2 - 26.8984 x2^2 + 25515.6 x3 - 2185.12 x1 x3 + 223.281 x2 x3 + 11075. x3^2 + 459.875 x4 + 9.635 x1 x4 - 44.2687 x2 x4 - 6. x3 x4 - 18.28 x4^2
SONR	(5.28854 + 89.4075 x1 + 1141.18 x1^2 + 68.6839 x2 + 832.775 x1 x2 + 649.404 x2^2 + 0.0360891 x3 + 11.298 x1 x3 + 7.35784 x2 x3 + 0.283969 x3^2 - 53.3765 x4 + 118.769 x1 x4 - 33.8006 x2 x4 - 28.4194 x3 x4 - 1029.25 x4^2)/(83.5656 + 830.828 x1 + 19.1435 x1^2 + 5.09064 x2 - 255.251 x1 x2 + 179.994 x2^2 + 73.7283 x3 - 485.302 x1 x3 - 212.805 x2 x3 - 12.0177 x3^2 - 1083.41 x4 + 75.2429 x1 x4 + 15.6012 x2 x4 + 481.127 x3 x4 + 18.5534 x4^2)
FOTN	-4995.88 + 1131.44 Cos[x1] + 81.3643 Cos[x2] + 5555.52 Cos[x3] - 46.211 Cos[x4] - 536.112 Sin[x1] + 133.206 Sin[x2] + 17526. Sin[x3] - 20.2339 Sin[x4]
FOTNR	(-3.97133*10^6 - 8.407*10^6 Cos[x1] - 3.7267*10^6 Cos[x2] - 3.7134*10^6 Cos[x3] - 9.00706*10^6 Cos[x4] - 4.53562*10^6 Sin[x1] + 61172.8 Sin[x2] - 1.43369*10^6 Sin[x3] + 2.44973*10^6 Sin[x4])/(-2.16583*10^7 + 1.75104*10^7 Cos[x1] +

	2.29055*10^7 Cos[x2] - 2.15529*10^7 Cos[x3] + 1.28199*10^7 Cos[x4] - 524641. Sin[x1] - 3.20662*10^7 Sin[x2] - 4.35674*10^6 Sin[x3] + 2.84404*10^6 Sin[x4])
SOTN	92.076 + 644.413 Cos[x1] - 665.792 Cos[x1]^2 + 103.046 Cos[x2] + 92.8243 Cos[x2]^2 + 109.206 Cos[x3] + 112.143 Cos[x3]^2 + 0.427725 Cos[x4] + 193.292 Cos[x4]^2 - 501.108 Sin[x1] + 234.38 Sin[x1]^2 + 114.619 Sin[x2] + 145.62 Sin[x2]^2 + 17568.6 Sin[x3] - 2788.12 Sin[x3]^2 - 23.586 Sin[x4] + 107.089 Sin[x4]^2
SOTNR	(-77401.3 + 26060.1 Cos[x1] - 33828.5 Cos[x1]^2 - 95278. Cos[x2] - 111619. Cos[x2]^2 - 77571. Cos[x3] - 77700.5 Cos[x3]^2 + 68191.7 Cos[x4] - 56542.3 Cos[x4]^2 + 32479.9 Sin[x1] - 43571.8 Sin[x1]^2 + 111536. Sin[x2] + 34218.7 Sin[x2]^2 - 5973.12 Sin[x3] + 300.142 Sin[x3]^2 + 34561.9 Sin[x4] - 20858. Sin[x4]^2)/(-15567.4 - 7194.87 Cos[x1] + 270.526 Cos[x1]^2 - 21126. Cos[x2] - 8692.7 Cos[x2]^2 + 11873. Cos[x3] + 35410.3 Cos[x3]^2 + 3979.24 Cos[x4] - 6480.17 Cos[x4]^2 + 15400.5 Sin[x1] - 15836.9 Sin[x1]^2 - 18930.6 Sin[x2] - 6873.69 Sin[x2]^2 + 36628.1 Sin[x3] - 50976.7 Sin[x3]^2 - 2731.33 Sin[x4] - 9086.23 Sin[x4]^2)
FOLN	12291.1 - 2177.96 Log[x1] + 355.048 Log[x2] + 2895.79 Log[x3] + 110.791 Log[x4]
FOLNR	(28028.8 + 69.1982 Log[x1] + 629.745 Log[x2] + 5315.24 Log[x3] + 57.1559 Log[x4])/(-3.48448 + 2.39472 Log[x1] - 0.178325 Log[x2] - 2.68364 Log[x3] + 0.0356056 Log[x4])
SOLN	19784.7 - 1836.43 Log[x1] + 621.671 Log[x1]^2 - 1750.96 Log[x2] - 297.06 Log[x2]^2 - 906.573 Log[x1 x2] + 2431.86 Log[x3] + 1336.38 Log[x3]^2 - 966.008 Log[x1 x3] + 4938.29 Log[x2 x3] - 1645.84 Log[x4] + 572.085 Log[x4]^2 - 876.552 Log[x1 x4] - 856.008 Log[x2 x4] + 1203.13 Log[x3 x4]
SOLNR	(-62613.3 + 169181. Log[x1] + 910565. Log[x1]^2 - 388058. Log[x2] - 1.19078*10^6 Log[x2]^2 - 218878. Log[x1 x2] - 56883.4 Log[x3] + 288244. Log[x3]^2 + 112296. Log[x1 x3] - 444942. Log[x2 x3] - 11709.7 Log[x4] + 229447. Log[x4]^2 + 157470. Log[x1 x4] - 399769. Log[x2 x4] - 68594.2 Log[x3 x4])/(-6536.51 - 1803.26 Log[x1] + 1032.86 Log[x1]^2 + 3749.98 Log[x2] - 4674.99 Log[x2]^2 + 1945.72 Log[x1 x2] + 402.4 Log[x3] + 1220.27 Log[x3]^2 - 1401.86 Log[x1 x3] + 4151.38 Log[x2 x3] - 141.234 Log[x4] - 417.734 Log[x4]^2 - 1945.49 Log[x1 x4] + 3607.75 Log[x2 x4] + 260.166 Log[x3 x4])
H	277.631 - 110.818 x1 - 19.083 x1^2 + 111.219 x2 + 23.2752 x2^2 + 8861.08 x3 - 2134.5 x3^2 + 36.2057 x4 + 4.21497 x4^2 - 13524.7 Sin[x1] - 27.3716 Cos[x2] Sin[x1] + 13742.8 Cos[x3] Sin[x1] + 167.816 Cos[x4] Sin[x1] - 531.801 Sin[x1]^2 + 2999.71 Sin[x2] - 3911.14 Cos[x3] Sin[x2] -

	368.964 Cos[x4] Sin[x2] + 459.576 Sin[x2]^2 + 9244.6 Sin[x3] - 143.196 Cos[x4] Sin[x3] + 456.96 Sin[x3]^2 - 170.588 Sin[x4] + 287.886 Sin[x4]^2

Table 8. Full form mathematical models for linear thickness (Related to Table 4)

L	0.71522 + 0.00262113 x1 - 0.00110096 x2 + 0.00399257 x3 + 0.000302629 x4
LR	(6424.69 - 350.729 x1 - 220.269 x2 - 18773.7 x3 + 534.08 x4)/(8870.85 - 491.475 x1 - 296.943 x2 - 25679.6 x3 + 727.546 x4)
SON	0.681594 - 0.00451145 x1 + 0.000432234 x1^2 + 0.0170472 x2 - 0.000142 x1 x2 - 0.00168202 x2^2 - 0.0309312 x3 + 0.0197017 x1 x3 - 0.00082375 x2 x3 - 0.145554 x3^2 + 0.00403197 x4 - 0.00027776 x1 x4 + 0.000454615 x2 x4 - 0.007591 x3 x4 - 0.000219814 x4^2
SONR	1.0446 + 0.313529 x1 + 0.842216 x1^2 + 2.43377 x2 + 0.725672 x1 x2 + 0.656257 x2^2 + 1.03211 x3 + 1.09017 x1 x3 + 0.957645 x2 x3 + 0.98289 x3^2 - 0.385654 x4 + 0.768286 x1 x4 + 1.16984 x2 x4 + 0.631771 x3 x4 + 0.989831 x4^2)/(0.975753 + 1.51137 x1 + 0.936391 x1^2 - 0.0102155 x2 + 1.00685 x1 x2 + 1.50946 x2^2 + 0.977804 x3 + 0.93253 x1 x3 + 1.03718 x2 x3 + 1.01257 x3^2 + 2.07171 x4 + 1.14563 x1 x4 + 1.13556 x2 x4 + 1.27885 x3 x4 + 1.30521 x4^2)
FOTN	0.567201 - 0.0105753 Cos[x1] + 0.00416971 Cos[x2] + 0.15867 Cos[x3] - 0.000453084 Cos[x4] + 0.00264954 Sin[x1] - 0.00334137 Sin[x2] + 0.0391829 Sin[x3] + 0.00127612 Sin[x4]
FOTNR	(4.51971 - 2.16554 Cos[x1] + 1.67597 Cos[x2] + 4.66224 Cos[x3] - 2.07675 Cos[x4] - 1.03701 Sin[x1] - 0.672079 Sin[x2] + 1.67333 Sin[x3] + 0.922599 Sin[x4])/(6.4142 - 2.72476 Cos[x1] + 2.22719 Cos[x2] + 6.14877 Cos[x3] - 2.82448 Cos[x4] - 1.45908 Sin[x1] - 0.859773 Sin[x2] + 2.12949 Sin[x3] + 1.25614 Sin[x4])
SOTN	0.101216 + 0.0260418 Cos[x1] + 0.190432 Cos[x1]^2 + 0.0169426 Cos[x2] + 0.111765 Cos[x2]^2 + 0.104922 Cos[x3] + 0.108792 Cos[x3]^2 + 0.0319255 Cos[x4] + 0.183903 Cos[x4]^2 + 0.000017799 Sin[x1] + 0.122753 Sin[x1]^2 - 0.0142914 Sin[x2] + 0.142868 Sin[x2]^2 + 0.0396034 Sin[x3] + 0.0801706 Sin[x3]^2 - 0.00105099 Sin[x4] + 0.124058 Sin[x4]^2
SOTNR	(1.47012 - 2.56392 Cos[x1] + 2.77263 Cos[x1]^2 - 1.06712 Cos[x2] + 0.751662 Cos[x2]^2 + 3.06648 Cos[x3] + 4.43929 Cos[x3]^2 + 0.326786 Cos[x4] + 1.64855 Cos[x4]^2 - 6.28028 Sin[x1] - 0.302514 Sin[x1]^2 - 0.132674 Sin[x2] + 1.71846 Sin[x2]^2 + 2.97457 Sin[x3] - 1.96917 Sin[x3]^2 + 5.0326 Sin[x4] + 0.821566 Sin[x4]^2)/(4.21771 - 3.78297 Cos[x1] + 4.08472 Cos[x1]^2 - 1.43062 Cos[x2] + 1.82546 Cos[x2]^2 + 3.06354 Cos[x3] + 2.06802 Cos[x3]^2 - 0.0704316 Cos[x4] + 2.68228 Cos[x4]^2 - 8.66376 Sin[x1] + 1.13299 Sin[x1]^2 - 0.256789 Sin[x2] + 3.39226 Sin[x2]^2 - 0.0118233 Sin[x3] + 3.1497 Sin[x3]^2 + 6.92547 Sin[x4] + 2.53543 Sin[x4]^2)

FOLN	0.701008 + 0.0177726 Log[x1] - 0.0049933 Log[x2] + 0.000845078 Log[x3] + 0.00261696 Log[x4]
FOLNR	(16.283 - 19.9253 Log[x1] - 9.78662 Log[x2] + 11.6147 Log[x3] + 39.3988 Log[x4])/(22.8004 - 27.8155 Log[x1] - 13.2257 Log[x2] + 15.7716 Log[x3] + 53.9476 Log[x4])
SOLN	0.558673 + 0.0170657 Log[x1] + 0.0276777 Log[x1]^2 + 0.0393459 Log[x2] - 0.0626481 Log[x2]^2 + 0.013662 Log[x1 x2] - 0.0294786 Log[x3] - 0.00226278 Log[x3]^2 - 0.130754 Log[x1 x3] + 0.14297 Log[x2 x3] + 0.0245326 Log[x4] - 0.0153018 Log[x4]^2 + 0.0104655 Log[x1 x4] + 0.015783 Log[x2 x4] + 0.0105777 Log[x3 x4]
SOLNR	(-0.438269 - 0.322369 Log[x1] + 1.17153 Log[x1]^2 + 2.29937 Log[x2] - 0.671994 Log[x2]^2 + 0.977002 Log[x1 x2] + 0.55317 Log[x3] + 1.60738 Log[x3]^2 - 0.769199 Log[x1 x3] + 1.85254 Log[x2 x3] + 1.20105 Log[x4] + 1.53078 Log[x4]^2 - 0.121316 Log[x1 x4] + 2.50042 Log[x2 x4] + 0.754223 Log[x3 x4])/(2.20965 + 1.74689 Log[x1] - 0.353022 Log[x1]^2 + 0.0503287 Log[x2] + 1.86856 Log[x2]^2 + 0.797218 Log[x1 x2] + 0.684186 Log[x3] + 2.14482 Log[x3]^2 + 1.43108 Log[x1 x3] - 0.265485 Log[x2 x3] + 1.47284 Log[x4] + 2.2118 Log[x4]^2 + 2.21973 Log[x1 x4] + 0.523173 Log[x2 x4] + 1.15703 Log[x3 x4])
H	0.101686 + 0.00867342 x1 + 0.000708469 x1^2 + 0.0165168 x2 + 0.00223175 x2^2 - 0.00807968 x3 + 0.222682 x3^2 + 0.00834084 x4 + 0.000476501 x4^2 - 0.251894 Sin[x1] + 0.00596318 Cos[x2] Sin[x1] + 0.239237 Cos[x3] Sin[x1] - 0.0137572 Cos[x4] Sin[x1] + 0.123966 Sin[x1]^2 - 0.0190306 Sin[x2] - 0.0786376 Cos[x3] Sin[x2] - 0.0371437 Cos[x4] Sin[x2] + 0.0387822 Sin[x2]^2 - 0.012294 Sin[x3] - 0.290306 Cos[x4] Sin[x3] + 0.145279 Sin[x3]^2 - 0.0120116 Sin[x4] + 0.174433 Sin[x4]^2