Comparative Study of LS-SVM, RVM and ELM for Modelling of Electro-Discharge Coating Process

Morteza Taheri*
Department of Mechanical Engineering,
Birjand University of Technology, Birjand, Iran
E-mail: morteza.taheri72@yahoo.com
*Corresponding author

Nader Mollayi
Department of Computer Engineering,
Birjand University of Technology, Birjand, Iran
E-mail: mollayi@birjandut.ac.ir

Seyyed Amin Seyyedbarzani, Abolfazl Foorginejad
Department of Mechanical Engineering,
Birjand University of Technology, Birjand, Iran
E-mail: barzani.amin@gmail.com, foorginejad@birjandut.ac.ir

Vahide Babaiyan
Department of Computer Engineering,
Birjand University of Technology, Birjand, Iran
E-mail: babaiyan@birjandut.ac.ir

Received: 11 April 2020, Revised: 19 August 2020, Accepted: 1 October 2020

Abstract: The Electro-discharge coating process is an efficient method for improvement of the surface quality of the parts used in molds. In this process, Material Transfer Rate (MTR), an average Layer Thickness (LT) are important factors, and tuning the input process parameters to obtain the desired value of them is a crucial issue. Due to the wide range of the input parameters and nonlinearity of this system, the establishment of a mathematical model is a complicated mathematical problem. Although many efforts have been made to model this process, research is still ongoing to improve the modeling of this process. To this end, in the present study, three powerful machine learning algorithms, namely, Relevance Vector Machine (RVM), Extreme Learning Machine (ELM) and the Least Squares Support Vector Machine (LS-SVM) that have not been used to model this process, have been used. The values R^2 above 0.99 for the training data and above 0.97 for the test data show the high accuracy and generalization capability degree related to the LS-SVM models, which can be applied for the input parameters tuning in order to attain a preferred value of the outputs.

Keywords: Average Layer Thickness, Electro-Discharge Coating, Material Removal Rate, Support Vector Machine

Reference: Morteza Taheri, Nader Mollayi, Seyed Amin Seyyedbarzani, Abolfazl Foorginejad, and Vahide Babaiyan, "Comparative Study of LS-SVM, RVM and ELM for Modelling of Electro-Discharge Coating Process", Int J of Advanced Design and Manufacturing Technology, Vol. 14/No. 1, 2021, pp. 19–32. DOI: 10.30495/admt.2021.1897266.1190

Biographical notes: Morteza Taheri is PhD Student of Mechanical Engineering, Energy Conversion at Tarbait Modarres University, Iran. He received his BSc in Mechanical Engineering from Birjand University of Technology, Iran. **Nader Mollayi** received his MSc in Eelctrical Engineering from Sharif University of Technology in 2009. He is currently a lecturer at the Department of Computer Engineering and information Technology in Birjand University of Technology, Iran. His fields of interest are signal processing, soft computing and electronic system design. **Seyed Amin Seyyedbarzani** received his MSc in Mechanical Engineering from Birjand University of Technology. His fields of interest are Rapid Prototyping and Reverse Engineering. **Abolfazl Foorginejad** is currently Assistant Professor at the Department of Mechanical Engineering, Birjand University of Technology, Birjand, Iran. His current research interest includes Rapid Prototyping and Reverse Engineering. **Vahide Babaiyan** is currently a lecturer at the Department of Computer Engineering in Birjand University of Technology, Birjand, Iran. Her research interests include IoT, WSN, data mining and machine learning.

1 INTRODUCTION

Electro-Discharge Machining (EDM) is a non-traditional electrical-thermal machining process which is operated by precise control of sparks between an electrode and a workpiece in presence of a dielectric liquid, while the electrode is considered as the cutting tool. This process is widely used for machining hard materials [1]. In this process, contrary to other machining methods, there is no physical contact between tool and workpiece for removal of the material and therefore there is no tool force. Hence, the EDM process has found widespread application in molding industry during the recent decades [2]. The main disadvantages of this process are tool wear and the formation of a thin brittle coating on the machined surfaces [3]. The tool wear is controllable to some extent. However, achieving a condition with no tool wear seems to be impossible. Scholars have introduced a novel method which makes it possible to improve the workpiece surface quality with the use of electro-discharge process and hydrocarbon dielectric liquid. In this method which is referred to as Electro-Discharge Coating (EDC), the tool is made by powder metallurgy and a hard carbide surface is formed on the workpiece in presence of a dielectric fluid of hydrocarbons [4-5]. The schematic view of EDC process is depicted in "Fig. 1" [6]. The EDC process begins with the wear of tool which follows the formation of hard carbides via a chemical reaction between the materials produced from corrosion of electrode and carbon particles generated from decomposition of hydrocarbon in high temperature, and the carbide coating is created on the workpiece surface in a few minutes. In this process, the properties of powder metallurgy tool can be determined by controlling the concentration pressure and sintering temperature [7-8].

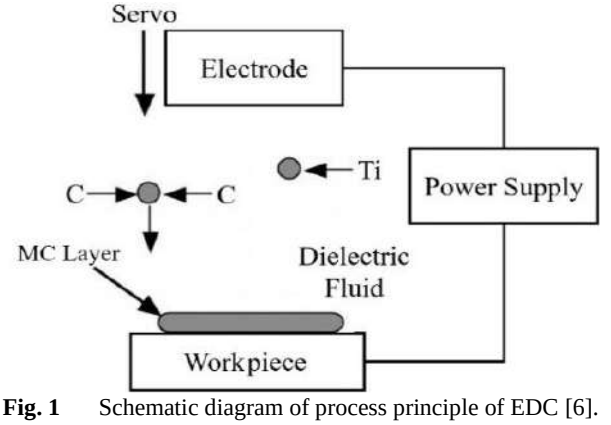

Fig. 1 Schematic diagram of process principle of EDC [6].

In the EDC process, the Material Transfer Rate (MTR) as an index of the processing speed, and average Layer Thickness (LT) as an index of the surface quality, are important factors, which are influenced by input process parameters such as Concentration Pressure (CP), Sintering Temperature (ST), Electric Current Intensity (lp), Pulse-on Time (Ton), Pulse-off Time (Toff), and appropriate choosing of these parameters to obtain a preferred value of MTR and LT is considered as a critical issue in this process. For this goal, it is very important to establish a mathematical predictive model for these factors globally, in terms of the input process parameters. On the other hand, the EDC process is a highly nonlinear coupled multivariable system with a complex and stochastic nature, in which a small change in one variable can abruptly change the output [9]. Moreover, the output parameters are also in a wide numerical range. Therefore, the establishment of the predictive models is a complicated mathematical problem. In recent decades, advances in machine learning have provided the possibility of solving many practical complex problems easier. Regarding this field of computer science, investigation and construction of algorithms, which have the capability of learning, and predicting with respect to a limited set of obtained data, is discovered [1], [4-10]. A model, in these algorithms, is constructed from example inputs to create data-driven predictions or decisions. Supervised learning is the learning task machine that infers a function from a labeled training data set [11], and it is possible to develop a predictive model of a function based on supervised learning algorithms, with respect to a limited number of observations. For example, Tyagi et al. [12] investigated the effect of duty factor, peak current, and powder mixing ratio, on tool wear rate, mass transfer rate, microhardness, and coating thickness. Also, they applied an artificial neural network for the prediction of output parameters response. The results showed good agreement between experimental and predicted data using the Artificial Neural Network (ANN). Sahu et al. [13] investigated in the EDC process, the effect of process parameters on the surface finish properties of the coated specimen. Their goal was to reduce the number of experiments, optimize the surface properties of the specimen by Taguchi based VIKOR method combined with Harmony search algorithm and the best parametric setting is selected. In this regard, in this research, an attempt has been made to use powerful machine learning algorithms that have not been used to improve the EDC process in order to improve process prediction. In a scholar by Patowari et al. [9], for modeling and predicting the MTR and LT in an EDC process by applying the Artificial Neural Networks (ANN), as a basic supervised learning algorithm and also by using a worldwide database of process parameters, has been recommended. The database was attained, with respect to the experiments carried out by a Victor-I EDM set prepared by Electronica Machine Tools Ltd, Pune. In these experiments, the electrode was made from powder metallurgy constituting 75% tungsten and 25% copper.

The difference between workpiece weights before and after the process was measured by an electronic scale for the determination of the MTR value. For measurement of the LT, the workpiece was sectioned by wire cut and after the preparation of the sectioned surface, the layer thickness was measured by an optical micrometer. The input process parameters and the corresponding values of MTR and LT are listed in "Table 1".

Today, scholars have presented novel machine learning algorithms which are of valuable advantage over the traditional artificial neural networks, and regarding the complex nature of the EDC process, a wide numerical range of the parameters and importance of modelling accuracy and application of these algorithms in this problem can be beneficial to obtain a higher degree of accuracy. In this paper, this problem has been investigated based on three powerful machine learning algorithms, namely, Relevance Vector Machine (RVM), Extreme Learning Machine (ELM), and the Least Squares Support Vector Machine (LS-SVM). Based on the modeling results, the LS-SVM models benefit from a greater accuracy and generalization capability degree in these method, and also compared to the ANN-based method.

Table 1 Input parameters of the EDC process and the corresponding values of MTR and LT [9]

Expt. no.	Compt. pressure (CP), MPa	Sintering temp. (ST), °C	Peak current (Ip), A	On time (Ton), µs	Off time (Toff), µs	MTR, mg/min	LT, µm
1	120	700	8	19	19	4.91	30.4
2	180	700	8	19	19	5.05	23.9
3	240	700	8	19	19	3.04	24.5
4	300	700	8	19	19	0.93	16.7
5	120	700	8	28	29	6.30	24.9
6	180	700	8	28	29	8.52	31.1
7	240	700	8	28	29	7.13	24.3
8	300	700	8	28	29	3.12	22.8
9	120	700	8	38	39	5.43	33.8
10	180	700	8	38	39	9.46	40.2
11	240	700	8	38	39	6.86	36.2
12	300	700	8	38	39	1.11	14.5
13	120	700	8	58	59	8.14	41.4
14	180	700	8	58	59	10.86	30.2
15	240	700	8	58	59	4.92	26.1
16	300	700	8	58	59	6.02	37.4
17	120	700	8	126	54	56.37	229.6
18	180	700	8	126	54	38.27	166
19	240	700	8	126	54	22.54	105.7
20	300	700	8	126	54	13.61	67.2
21	120	700	8	256	108	127.73	521.9
22	180	700	8	256	108	87.74	318.5
23	240	700	8	256	108	60.4	206.5
24	300	700	8	256	108	27.94	123.3
25	120	700	8	386	162	172.56	739
26	180	700	8	386	162	107.46	446.2
27	240	700	8	386	162	104.59	437.3
28	300	700	8	386	162	68.57	229.9
29	120	700	4	126	54	7.95	30.4
30	180	700	4	126	54	6.08	25
31	240	700	4	126	54	6.17	24.4
32	300	700	4	126	54	6.14	24
33	120	700	4	256	108	11.46	36
34	180	700	4	256	108	7.4	24.7
35	240	700	4	256	108	6.6	18.9
36	300	700	4	256	108	2.45	19.6
37	120	700	4	386	162	58.54	223.4
38	180	700	4	386	162	34.18	110.4
39	240	700	4	386	162	27.72	112.4
40	300	700	4	386	162	20.41	75
41	120	700	12	19	19	3.94	32.5
42	180	700	12	19	19	3.2	15.6
43	240	700	12	19	19	1.47	9.3

44	300	700	12	19	19	-0.4	14.6
45	120	700	12	28	29	7.61	34.2
46	180	700	12	28	29	5.41	39.1
47	240	700	12	28	29	0.24	15.1
48	300	700	12	28	29	-0.47	14.3
49	120	700	12	38	39	5.39	29.2
50	180	700	12	38	39	6.34	32.1
51	240	700	12	38	39	1.77	23.5
52	300	700	12	38	39	0.1	19.1
53	120	700	12	58	59	9.87	41.6
54	180	700	12	58	59	3.73	32
55	240	700	12	58	59	1.25	19.5
56	300	700	12	58	59	0.37	6.2
57	120	700	6	126	54	51.76	183.6
58	120	700	6	256	108	89.95	348
59	120	700	10	126	54	102.91	420.1
60	120	700	10	256	108	190.84	785.2
61	120	900	8	19	19	4.98	31.5
62	180	900	8	19	19	1.77	12.3
63	240	900	8	19	19	1.4	12.6
64	300	900	8	19	19	1.08	13.6
65	120	900	8	28	29	4.85	35.2
66	180	900	8	28	29	1.38	13.3
67	240	900	8	28	29	0.27	8.2
68	300	900	8	28	29	-0.5	8.8
69	120	900	8	38	39	6.65	29.9
70	180	900	8	38	39	0.78	9
71	240	900	8	38	39	-0.76	3.1
72	300	900	8	38	39	-1.52	16.8
73	120	900	8	58	59	5.77	34.4
74	180	900	8	58	59	0.89	13
75	240	900	8	58	59	-1.52	13.4
76	300	900	8	58	59	-5.08	17.4
77	120	900	8	126	54	6.17	35.2
78	180	900	8	126	54	-1.11	20.9
79	240	900	8	126	54	-5.07	19.2
80	300	900	8	126	54	-5.77	21
81	120	900	12	19	19	13.16	81.6
82	180	900	12	19	19	6.14	33.6
83	240	900	12	19	19	1.23	14.7
84	300	900	12	19	19	0.37	10.6
85	120	900	12	28	29	16.94	69.5
86	180	900	12	28	29	2.57	15.7
87	240	900	12	28	29	-3.13	6.8
88	300	900	12	28	29	-1.65	15.7
89	120	900	12	38	39	16.98	71.3
90	180	900	12	38	39	5.36	31.6
91	240	900	12	38	39	-4.29	7.5
92	300	900	12	38	39	-3.09	12.6
93	120	900	12	58	59	12.51	72.6
94	180	900	12	58	59	2.01	18.8
95	240	900	12	58	59	-3.14	11.4
96	300	900	12	58	59	-0.53	13.2
97	120	900	12	126	54	15.09	55.2
98	180	900	12	126	54	-5.67	26.8
99	240	900	12	126	54	-13.84	25.2
100	300	900	12	126	54	-11.89	22.2
101	120	700	12	126	54	74.47	-
102	300	700	12	126	54	21.71	-

2 SUPPORT VECTOR MACHINES

Support Vector Machines (SVMs) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. The initial form of the support vector machines was a generalization of the Generalized Portrait algorithm developed in the 1960s [14]. However, the present form of the SVMs was developed by Vapnik and his coworkers at the AT&T Bell Laboratories in the 1990s [15]. Reducing the Structural risk as well as reducing the empirical risk is the key benefit of the SVMs compared to the neural networks that, in many practical applications, leads to a better generalization capability [16].

As shown in "Fig. 2", In SVM-based categorization, a hyper-plane is obtained, which results in an equivalent maximum margin between these two classes samples in the training dataset. Accordingly, such a classifier can be explained by "Eq. (1)", where w and b are the weights and bias vector, respectively.

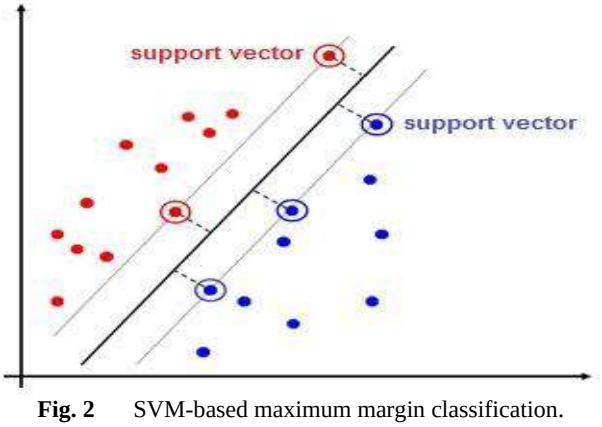

Fig. 2 SVM-based maximum margin classification.

$$f(x) = sign(\, w^T\, x + b\,) \tag{1}$$

Increasing the margin between these two classes is accomplished through reducing the risk function R(w) that is expressed in "Eq. (2)", exposed to the "Eq. (3)" constraints, for the N samples of the (x_i, y_i) in the training dataset:

$$R(w) = \frac{1}{2} w^T w = \frac{1}{2} \|w\|^2 \tag{2}$$

$$y_i\, (w^T\, x_i + b) \geq 1, \;\; i = 1, \dots, N \tag{3}$$

The closest samples to the hyper-plane are the support vectors, as depicted in "Fig. 2". The support vectors lay on a hyper-plane satisfying the condition of:

$$y_{sp}\, \left(w^T\, x_{sp} + b\right) = 1 \tag{4}$$

If the samples of two classes in the training database are not linearly separable, another factor must be added to the risk function in "Eq. (2)" for the inevitable error in the case of the samples which lay outside the permitted borders. Therefore, the optimization problem is converted for reducing the risk function that is described in "Eq. (5)" exposed to the "Eq. (6)" restrictions. As shown in "Fig. 3", in "Eq. (5)", ξ_i stands for the distance of the support vectors' hyper-plane from the samples that lay outside it. Parameter C is identified as the regularization factor, which trades off the associated importance of enhancing the margin and training error- i.e., the structural and empirical risks.

$$\mathrm{R}(w) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{N} \xi_i \tag{5}$$

$$y_i\, (w^T\, x_i + b) \geq 1 - \xi_i, i = 1, \dots, N \tag{6}$$

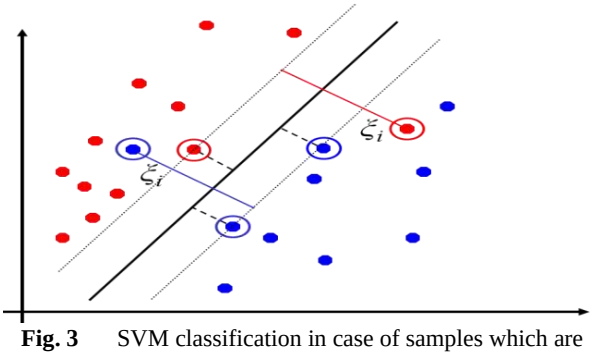

Fig. 3 SVM classification in case of samples which are not linearly separable.

This method, however useful, does not lead to acceptable classification error in the case of feature space, in which the orientation of the border between the two classes' samples is non-linear. As shown in "Fig. 4", in such a condition, the original feature space can be mapped to some feature spaces that have higher dimensional where the training set can be separated, via a nonlinear function considered as the kernel function [17].

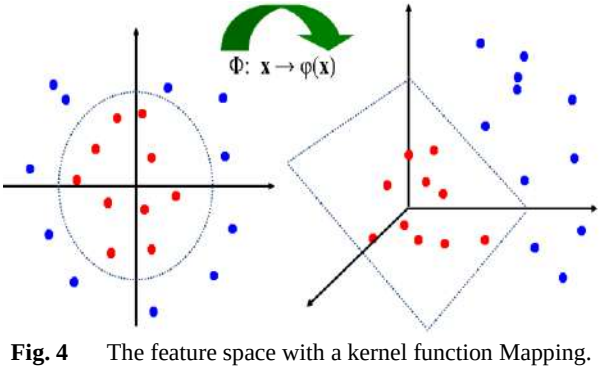

Fig. 4 The feature space with a kernel function Mapping.

In this case, the SVM-based classifier system can be explained as:

$$f(x) = sign(\, w^T \, \phi(x) + b \,) \tag{7}$$

Where, w and b are attained by reducing the risk function R (w) in "Eq. (5)" exposed to the constraints of:

$$y_i \, (w^T \, \phi(x_i) + b) \geq 1 - \, \xi_i, \tag{8}$$

In the training database, for the N samples, the regression problem can be described as a function approximating according to a limited number of observations of cases that were not observed. A margin of tolerance is usually considered for approximating a function in practical applications. For example, if the function to be approximated is expressed in Euro's, the permitted margin of tolerance will be 0.01 in order to consider the Eurocent, as shown in "Fig. 5". Due to a limited number of observations on the function f(x), along with the tolerance ε permitted margin and also a few numbers of its values, SVM-based categorization between f(x) + ε and f(x) - ε can be considered as approximating f(x) in the tolerance permitted margin [18]. Consequently, formulating support vector machines can be generalized in order to perform regression as:

$$\begin{aligned} y = f(x) = \sum_{i=1}^{m} w_i \phi_i(x) + b_i \\ = \; w^T \phi(x) + b \end{aligned} \tag{9}$$

The purpose is to compute the w and b values, with respect to a set of available training data, therefore the difference is reduced between the original function and the estimated function.

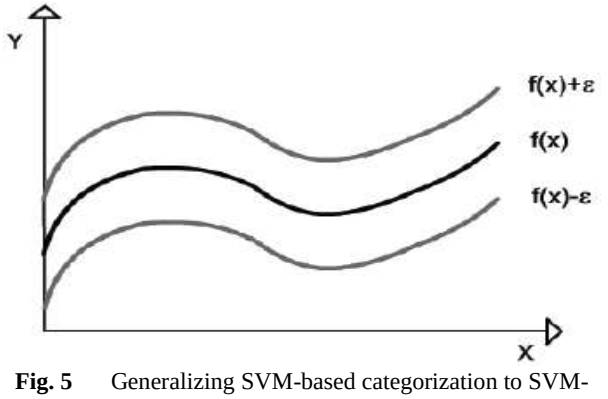

Fig. 5 Generalizing SVM-based categorization to SVM-based regression [18].

3 LEAST SQUARES SUPPORT VECTOR MACHINES

In classical SVMs, the optimization problem is solved via quadratic programming [19]. However, this method has a high computational burden shortcoming to plan constrained optimization. This weakness has been overcome using LS-SVMs that solve a linear equation set as a substitute for a quadratic programming difficulty [20].

The LS-SVMs optimization problem to estimate the function is expressed as:

$$\begin{aligned} \min \quad & \mathrm{R}(w) = \tfrac{1}{2}\|w\|^2 \; + \tfrac{C}{2}\textstyle\sum_{i=1}^{N} e_i^2, \\ \text{s.t.} \quad & y_i = \langle w, \varphi(x_i)\rangle + b + e_i \,, (\mathrm{i} = \\ & 1, \dots \,, N) \end{aligned} \tag{10}$$

Where, e_i is the regression error of the i-th training data, as shown in "Fig. 6", C≥0 is the regularization constant and N is the number of the training samples. This optimization problem can be regarded as a typical convex optimization problem that can be solved by the application of the Lagrange multipliers method. The Lagrangian is expressed by "Eq. (11)", with the Lagrange multipliers, $\alpha_i \in R$.

$$\begin{aligned} L_\alpha(w, b, e) = \frac{1}{2}\|w\|^2 \; + \frac{C}{2}\sum_{i=1}^{N} e_i^2 \\ - \sum_{i=1}^{N} \alpha_i \{\langle w, \varphi(x_i)\rangle + b + e_i - y_i\} \end{aligned} \tag{11}$$

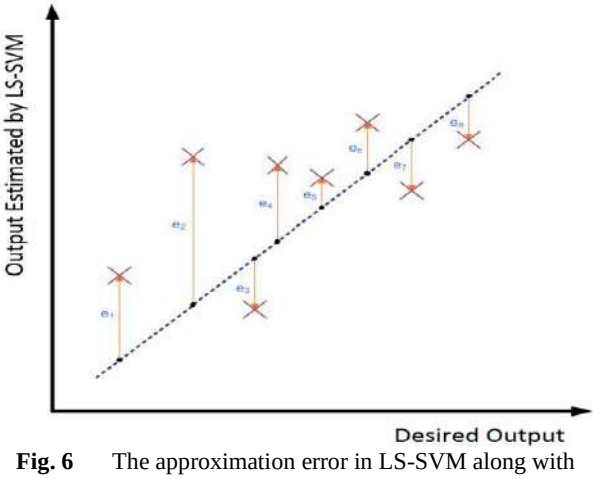

Fig. 6 The approximation error in LS-SVM along with eight samples of the training data set [20].

Differentiating the Lagrangian, the optimality conditions are obtained as:

$$\begin{aligned} \frac{\partial L_\alpha(w, b, e)}{\partial w} = 0 \; \rightarrow w = \sum_{i=1}^{N} \alpha_i \varphi(x_i) \\ \frac{\partial L_\alpha(w, b, e)}{\partial b} = 0 \; \rightarrow \sum_{i=1}^{N} \alpha_i = 0 \end{aligned} \tag{12}$$

$$\frac{\partial L_\alpha(w,b,e)}{\partial e_i} = 0 \rightarrow \alpha_i = C\ .\ e_i\ , (i = 1\ , \dots , N)$$
$$\frac{\partial L_\alpha(w,b,e)}{\partial \alpha_i} = 0 \rightarrow \langle w, \varphi(x_i)\rangle + b + e_i - y_i = 0$$

Replacement of w and e in the Lagrangian results in the subsequent linear Karush–Kuhn–Tucker (KKT) system:

$$\begin{bmatrix} 0 & 1_v^T \\ 1_v & \Omega + {}^{I}/_{C} \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \quad (13)$$

In this equation, I stands the size N identity matrix, and $y = [y_1, \dots, y_N]^T$, $1_v = [1, \dots, 1]^T$; $\alpha = [\alpha_1, \dots, \alpha_N]^T$ are N by 1 vector. Furthermore, the element in row k and column i of Ω is computed in terms of the subsequent equation [21].

$$\Omega_{ki} = \langle \varphi(x_k), \varphi(x_i)\rangle \quad (k\ , i = 1, \dots, N) \quad (14)$$

According to Mercer's theorem [17], the inner product $\langle \varphi(x), \varphi(x_i)\rangle$ can be defined through a kernel K(x, xi), so Ω_{ki} can be expressed as "Eq. (16)". The most frequent formulations for the kernel function are tabulated in "Table 2".

$$\Omega_{ki} = \langle \varphi(x_k), \varphi(x_i)\rangle = K(x_k, x_i), (\ k, i = 1, \dots, N) \quad (15)$$

The Gaussian radial basis (RBF) function is mainly preferred in regression problems. Estimating the LS-SVM-based function as well as the RBF kernel is described in "Eq. (14)", where σ^2 stands for the kernel parameter, and α_i and b are the solutions to "Eq. (11)".

$$f(x) = \sum_{i=1}^{N} \alpha_i K(x, x_i) + b = \sum_{i=1}^{N} \alpha_i exp\ (-\frac{\|x - x_i\|^2}{2\sigma^2}) + b \quad (16)$$

Table 2 Most common formulations for the kernel function

Kernel type	Formulation
Linear	$K(x, x_i) = \langle x, x_i\rangle$
Gaussian radial basis (RBF)	$K(x, x_i) = exp\ (-\frac{\|x - x_i\|^2}{2\sigma^2})$
Polynomial of degree d	$K(x, x_i) = (\langle x, x_i\rangle + p)^d, d \in N\ , p > 0$
Multi-Layer Perceptron (MLP)	$K(x, x_i) = tanh(k.\langle x, x_i\rangle + \theta), k, \theta > 0$

4 EXTREME LEARNING MACHINE

ELM is a machine learning method used in classification and regression for single-layer neural networks. In the learning process of this type of neural network, the number of nodes in the hidden layer can be set and randomly determines the weight of the inputs and hidden layers. The output layer weight is determined using the least square method. The learning process evolves through math changes without repetition [22]. This algorithm has a very fast learning speed. Experimental results show that this algorithm can often produce a good overall performance and act thousands of times faster than popular learning algorithms for feed-forward neural networks in the learning process [23].

In practical applications, the model should be trained based on a set of existing observations and then used in the prediction phase. Through iteration, in order to complete the learning process during the training period, the influential factors and related outcomes are put in the ELM model. This algorithm only requires a number of hidden layer nodes and in the implementation phase, the algorithm does not need to adjust the weight of network inputs and hidden biases.

For N arbitrary distinct samples (x_i , t_i), that $x_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T \in R^n$ and $t_i = [t_{i1}, t_{i2}, \dots, t_{in}]^T \in R^m$, what's more ($x_i$, t_i) $\in R^n \times R^m$ (i = 1, 2, . . . , N), standard single hidden layer feed forward networks (SLFN) with $\tilde{N}$ hidden nodes and f(x) as an activation function, mathematically are modeled as below [22-24]:

$$\sum_{i=1}^{\tilde{N}} \beta_i f_i(x_j) = \sum_{i=1}^{\tilde{N}} \beta_i f(a_i . x_j + b_i) = t_j\ , j = 1, \dots, N \quad (17)$$

Here $a_i = [a_{i1}, a_{i2}, \dots, a_{in}]^T$ is the weight vector relating the ith hidden node with the input nodes, and b_i is the threshold of the ith hidden node. $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T$ is the weight vector connecting the ith hidden node and the output nodes $a_i . x_j$ Indicates the inner product of a_i and x_i, and the activation function usually chooses "Sigmoid", "Sine", "RBF". The above "Eq. (17)" can be written succinctly as:

$$H\beta = T$$
$$H(a_1, \dots, a_{\tilde{N}}, b_1, \dots, b_{\tilde{N}}, x_1, \dots x_N) = \begin{bmatrix} f(a_1 . x_1 + b_1) & \cdots & f(a_{\tilde{N}} . x_1 + b_{\tilde{N}}) \\ \vdots & \ddots & \vdots \\ f(a_1 . x_N + b_1) & \cdots & f(a_{\tilde{N}} . x_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}}$$
$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{N \times m}, T = \begin{bmatrix} t_1^T \\ \vdots \\ t_{\tilde{N}}^T \end{bmatrix}_{N \times m} \quad (18)$$

H is named the hidden layer output matrix of the neural network; the ith column of H is the ith hidden node output according to inputs $x_1, x_2, \ldots, x_N$.

It can be proven that network parameters do not need to be set up when the number of hidden nodes is sufficient, and the activation function f(x) is an infinite distinction in each interval [25]. At the start of the training phase, the SLFN randomly appoint input connection weights *a* and hidden layer node biases *b*. In addition, while the training process remains unchanged, any continuous function can be estimated approximately. In general, to obtain a good disposition, take $\widetilde{N} \ll N$.

After the input weights and hidden layer biases are specified by random assignment, the hidden layer of the output matrix H can be obtained using the input samples. So, SFLN training turns into solving linear equations Hβ = T least-squares solution.

$$\begin{aligned}&\left\|H(a_1,\ldots,a_{\widetilde{N}},b_1,\ldots,b_{\widetilde{N}})\hat{\beta}-T\right\|\\&=\min_{\beta}\|H(a_1,\ldots,a_{\widetilde{N}},b_1,\ldots,b_{\widetilde{N}})\beta-T\|\end{aligned} \tag{19}$$

"Eq. (19)" least-squares solution of the above liner system is:

$$\hat{\beta} = H^{\dagger}T \tag{20}$$

In "Eq. (20)", $H^{\dagger}$ stands for Moore-Penrose [26], which is generalized in opposition of the matrix H hidden layer output. Generally, the optimal solution $\hat{\beta}$ comprises the next characteristics:

(1) According to $\hat{\beta}$, the least training error is attained by the algorithm;

(2) The minimum Pattern capability in optimal generalization, which is related to the output connection weights and the network could be performed.

(3) $\hat{\beta}$ is unique so that we do not need a locally optimal solution.

In short, we can say, given a training set $(x_i, t_i) \in R^n \times R^m$ (i = 1, 2, . . . , N), the activation function f (x) and hidden node number $\widetilde{N}$, and after that the ELM algorithm as followings at 3 stages:

Stage 1: Describing the hidden layer node number $\widetilde{N}$, randomly dedicate input weights a_i and concealed layer biasesb_i, (i = 1, 2,... , $\widetilde{N}$).

Stage 2: calculating the matrix H hidden layer output.

Stage 3: in terms of Eq. (20), determining the output weight β.

The benefits of the ELM algorithm are significant. Without iterative gradient-based training, many of the limitations of conventional algorithms based on regular gradients such as local minima, overtraining, and high computing are avoided. For each active infinite differentiation function, the ELM with hidden layer neurons can learn distinct samples with exactly zero error. Also, ELM training can always guarantee the best results with respect to the designated input weight In addition, ELM training can always guarantee the best results according to the assigned input weights. ELM also distinguishes it from traditional NNs in superior generalization capability without the overtraining issue [27].

5 RELEVANCE VECTOR MACHINE (RVM)

RVM is a specific case of a sparse kernel model, which indicates a Bayesian treatment of a generalized linear model of identical functional form to Support Vector Machine (SVM). RVM workflow and solution are different from SVM, which provides a possible interpretation of its output. This algorithm reduces complexity by creating models with a parametric structure and process which, together, is suitable for the content of data information [28].

Initially, the RVM has been derived and tested based on binary classification where it was expected to predict the membership in one of the classes given the input x. This statistical convention follows and promotes the generalization of the linear model. It follows the statistical convention and generalizes the linear model using the sigmoid logistic function $\sigma(y) = 1/(1 + e^{-y})$ to y(x) and adopting the Bernoulli distribution for P(t|x), the probability is written as [29]:

$$\begin{aligned}P(t|x) = \prod_{n=1}^{N} \sigma\{y(X_n; W)\}^{t_n}[1 \\ - \sigma\{y(X_n; W)\}]^{1-t_n}\end{aligned} \tag{21}$$

Nevertheless, unlike the regression case, weight is not integrative analytically, and so the closed-form expression for either the weight posterior $P(w|t, \alpha)$ or the marginal likelihood $P(t|\alpha)$, with α a vector N + 1 hyper-parameters are denied. based on Laplace's method, The approximation approach is used as next [29-30]:

1. The most feasible maximum posterior weights (W_{MP}) are found for a constant value of α, due to the location of the posterior distribution mode. Since $P(w|t, \alpha) \propto P(t|w)\, P(w|\alpha)$, this is equal to find the maximum, over w, of:

$$\begin{aligned}&\log\{P(t|w)p(w|\alpha)\}\\&= \sum_{n=1}^{N}[t_n \log y_n + (1 - t_n) \times \log(1 - y_n)] - \frac{1}{2} w^{T} A W\end{aligned} \tag{22}$$

2. Laplace's method is known as an easily quadratic approximation to the log-posterior around its state. "Eq. (22)" is twice discriminated, and after that provides:

$$\nabla w \nabla w \log p(p(w|t,\alpha)|_{W_{MP}} = -(\varphi^{T} B \varphi + A) \qquad (23)$$

Where, B = diag ($\beta_1, \ldots, \beta_N$) is a diagonal matrix with $\beta_n = \sigma\{y(X_n)\}[1 \_ \sigma(X_n)]$.

3. The hyper-parameters is updated by the use of:

$$\alpha_i^{new} = \frac{\gamma_i}{\mu_i^2} \qquad (24)$$

Where, $\gamma_i \equiv 1 - \alpha_i \sum_{ii}$; $\sum_{ii}$ stands for the ith diagonal element of the covariance $\sum = (\varphi^{T} B \varphi + A)^{-1}$ and μ is as same as the $W_{MP} = \sum \varphi^{T} Bt$. during presenting an extension to the multi-class issue, the chief RVM formulation fundamentally handles the K multi-class problem as a series on n one-against-all binary categorization problem. Accordingly, this would be interpreted into independently training n binary classifiers. The possibility in eq. (21) can be generalized to regular multinomial form as following:

$$P(t|w) = \prod_{n=1}^{N} \prod_{k=1}^{K} \sigma\{y_K(X_n; W_k)\}^{t_n k} \qquad (25)$$

Where, $t_n k$ stands for the index variable of observing n to be in class k; y_k stands for the predictor of class k. At this point, a true multi-class probability can be described as $P(t|w) = \prod_{n=1}^{N} \prod_{k=1}^{K} \sigma\{y_K; y_1, y_2, \ldots, y_k\}^{t_n k}$, where each class of $y_k$predictors is coupled in the multinomial logic function.

$$\sigma(y_K; y_1, y_2, \ldots, y_k) = \frac{e^{y_k}}{(e^{y_1} + e^{y_2} + \cdots + e^{y_k})} \qquad (26)$$

5.1 RVM REGRESSION MODEL

The regression model of RVM begins with the idea of linear models, which are commonly utilized in a variety of regression issues. Given a data set of input target pair $\{x_n, t_n\}_{n=1}^{N}$, considering scalar-valued target functions only, we follow the standard probabilistic formulation and presume that the targets are samples from the model with increasable noise [31]:

$$t_n = y(x_n; \omega) + \epsilon_n \qquad (27)$$

Where, ϵ_n are independent samples from some noise process which is further considered to be mean-zero Gaussian with variance σ^2. So the notation determines a Gaussian distribution over $t_n$with mean $y(x_n)$ and variance σ^2. Given the assumption of independence of the t_n, the probability of the complete data set can be written as:

$$p(t|\omega, \sigma^2) = \prod_{i=1}^{N} N(t_i|\phi_i \omega_i, \sigma^2) = (2\pi\sigma^2)^{-N/2} exp\{-\frac{1}{2\sigma^2}||t - \phi\omega||^2\} \qquad (28)$$

Where, $= (t_1, \ldots t_N)^T, \omega = (\omega_1, \ldots \omega_N)^T, \phi^{N \times M} = [\phi_1, \ldots, \phi_M]$ is a general N×M design matrix with column vectors $\phi_m = [1, K(x_m, x_1), K(x_m, x_2), \ldots, K(x_m, x_m)]$ and which $K(x_m, x_1)$ is a kernel function.

With as many parameters in the model as training examples, we would look for maximum-likelihood calculation of ω and σ^2 from eq. (28) to induce considerable over-fitting. To avoid this, we should add the penalty to the likelihood; we encode a priority for less complex functions by making the popular choice of a zero-mean Gaussian prior distribution over ω:

$$p(\omega|\alpha) = \prod_{i=0}^{N} N(\omega_i|0, \alpha_i^{-1}) \qquad (29)$$

Here α is a vector of M hyperparameters. To full-out the characteristics of this hierarchical prior, we must explain hyper priors over α, as well as over the final remaining parameter in the model, the noise variance σ^2. The appropriate priors are Gamma distributions since it is the Gauss distribution variance reciprocal conjugate distribution.

$$p(\alpha) = \prod_{i=0}^{N} Gamma(\alpha_i|a, b) \qquad (30)$$

$$pp(\beta) = Gamma(\beta|c, d) \qquad (31)$$

With $\beta \equiv \sigma^2$ and where:

$$Gamma(\alpha|a, b) = \Gamma(a)^{-1} b^a \alpha^{a-1} e^{-ba} \qquad (32)$$

$$\Gamma(a) = \int_0^{+\infty} t^{a-1} e^{-t} dt \qquad (33)$$

Where, $\Gamma(a)$ is the gamma function and to make these priors non-informative, we might fix their parameters to small values, such as $a = b = c = d = 0$, and then:

$$p(\omega_i|0, \alpha_i^{-1}) Gamma(\alpha_i|a, b) d\alpha_i \qquad (34)$$

Since this prior distribution is Automatic Relevance Determination (ARD) prior distribution, the sample vectors matching to nonzero weights of the basic functions are named relevance vectors after the training is completed and this training model is named relevance vector machine. Having defined the prior, Bayesian inference is obtained by calculating the posterior on all unknowns according to the data:

$$p(\omega,\alpha,\sigma^2|t)=\frac{p(t|\omega,\alpha,\sigma^2)p(\omega,\alpha,\sigma^2)}{p(t)} \tag{35}$$

Nevertheless, we cannot work out the solution of the posterior immediately because we cannot carry out the normalizing integral $p(t)=\int p(t|\omega,\alpha,\sigma^2)p(\omega,\alpha,\sigma^2)d\omega d\alpha d\sigma^2$. Alternatively, we analyze the posterior and then the posterior distribution of weights is gained from Bayes rule:

$$p(\omega|t,\alpha,\sigma^2)=N(\omega|\mu,\Sigma) \tag{36}$$

Where the posterior covariance and mean are:

$$\Sigma=(A+\sigma^{-2}\phi^T\phi)^{-1} \tag{37}$$

$$\mu=\sigma^{-2}\Sigma\phi^T t \tag{38}$$

We have defined $A=diag(\alpha_0,\alpha_1,\ldots,\alpha_N)$. So, machine learning becomes a search for the hyperparameters posterior-most probable. Predictions for new data are constructed with respect to the integration of the weights to take the marginal likelihood for the hyperparameters, which can be identified as the type-II maximum likelihood method.

$$\begin{aligned}p(t|\alpha,\sigma^2)&=\int p(t|\omega,\sigma^2)(\omega|\alpha)d\omega\\&=(2\pi)^{-\frac{N}{2}}|\sigma^2 I+\\&\phi A^{-1}\phi^T|^{-\frac{1}{2}}\exp\{-\frac{1}{2}t^T(\sigma^2\\&+\phi A^{-1}\phi^T)^{-1}t\}\end{aligned} \tag{39}$$

For regression, the type-II maximum probability issue usually solved by three methods, which include MacKay iteration estimation, the expectation-maximization iteration estimation, and the fast marginal likelihood maximization method. The ultimate hyper-parameters upgrade outcomes of these three methods are as follows:
(1) MacKay iteration estimation [32]:

$$\alpha_i^{new}=\frac{\gamma_i}{\mu_i^2} \tag{40}$$

$$\gamma_i=1-\alpha_i\Sigma_{ii} \tag{41}$$

$$(\sigma^2)^{new}=\frac{||t-\phi\mu||^2}{N-\Sigma_i\gamma_i} \tag{42}$$

(2) The expectation maximization iteration estimation [33]:

$$\alpha_i^{new}=\frac{1+2a}{[\omega_i^2]+2b} \tag{43}$$

$$(\sigma^2)^{new}=||t-\phi\mu||^2+\frac{(\sigma^2)^{old}}{N}\Sigma_i\gamma_i \tag{44}$$

(3) Fast marginal likelihood maximization method [34]:

(a) If $Q_i^2>S_i$ and $\alpha_i<\infty$, then $\alpha_i^{new}=\frac{S_i^2}{Q_i^2-S_i}$

Where:

$Q_i\triangleq\phi_i^T c_{-i}^{-1}t, s_i\triangleq\phi_i^T c_{-i}^{-1}\phi_i$ and $C=\sigma^2 I+\phi A^{-1}\phi^T$.

(b) If $Q_i^2<S_i$ then $\alpha_i^{new}=\infty$.

6 RESULTS AND DISCUSSION

The database of measurements of the LT and MTR and the corresponding input parameters, listed in "Table 1" [9] was used to develop the ELM, RVM and, LS-SVM models. From this database, shown in "Table 1", thirty-three samples marked in bold, were used for evaluating the accuracy of the models, which were trained by the rest of the samples.
For purpose of normalization, we scaled all of the input and target values within the range of [-1 +1] as:

$$pn=2*\frac{p-\left(\frac{max+min}{2}\right)}{(max-min)} \tag{45}$$

Where, *max* and *min* stand for the maximum and minimum input or the output values in the entire dataset, respectively, p stands for the input or output and pn signifies the corresponding normalized value.
With respect to the normalized dataset, free-source MATLAB implemented the ELM and RVM models, and implementations were prepared by [35]. Moreover, LS-SVM models were o\ implemented using the LS-SVM lab toolbox version 1.8 for MATLAB. Also, The toolbox offers a function for the kernel parameters and regularization constant tuning [36]. The ELM and RVM models' best parameters were also obtained through a wide numerical search, as listed in "Table 3". In progress, based on the trained models, the outputs were predicted and scaled to their original range as:

$$\hat{y} = y_n * \left(\frac{max - min}{2}\right) + \left(\frac{max + min}{2}\right) \quad (46)$$

In which, $\hat{y}$ is the predicted output and y_n is its normalized value.

Table 3 The LS-SVM, ELM and, RVM parameters

Model	Parameter	MTR	LT
LS-SVM	RBF Kernel parameter (σ^2)	0.2	1
	Regularization factor (C)	10000	$6.061*10^4$
RVM	Kernel parameter	1.64	1.4
ELM	kernel parameter	9	40
	Regularization coefficient	1700000	1000000

For comparison purposes, the models' predictive accuracy was evaluated based on the normalized. Outputs using the Mean Square Error (MSE), are defined as:

$$MSE = \frac{\sum_{i=1}^{N}(y_{n_i} - \widehat{y_{n_i}})^2}{N} \quad (47)$$

In this equation, the measured and the predicted normalized outputs are signified by y_{n_i} and$\widehat{y_{n_i}}$, respectively, and N stands for the training samples number. The calculated MSE value together with the MSE of the ANN method proposed by [9] are listed in "Table 4". As it can be observed, the LS-SVM method benefits from better accuracy for both the test and training and therefore a better generalization capability than the ANN method.

For further evaluation of the LS-SVM models' accuracy, the coefficient of determination (R^2), defined as "Eq. (48)" is also calculated.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - M)^2} \quad (48)$$

In this equation, y_i and $\hat{y}_i$ denote the measured and the predicted outputs, respectively, and M stands for the measured outputs mean value. The computed values of R^2 are tabulated in "Table 5" indicating the high degree of LS-SVM models' precision. The predicted outputs alongside their corresponding target values are depicted in "Figs. 7, 8". It can be observed that despite the outputs' wide numerical range, the LS-SVM models are able to predict them within a satisfactory level of accuracy.

Table 4 Comparisons of various models' mean square error

Database	Method	MSE	
Output		MTR	LT
Training	LS-SVM	$8.875*10^{-5}$	$1.808*10^{-4}$
	ANN [9]	$1.4*10^{-3}$	$1.6*10^{-3}$
	ELM	$4.45 * 10^{-5}$	$8.78 * 10^{-4}$
	RVM	$5.05 * 10^{-4}$	$1.695 * 10^{-4}$
Testing	LS-SVM	$9.544* 10^{-4}$	$6.69*10^{-4}$
	ANN [9]	$3.8*10^{-3}$	$2*10^{-3}$
	ELM	$2.299 * 10^{-3}$	$3.04 * 10^{-3}$
	RVM	$2* 10^{-3}$	$2.286 * 10^{-3}$

Table 5 The LS-SVM predictive models' coefficient of determination (R^2)

Database	Output	R^2
Training	MTR	0.9983
	LT	0.9943
Testing	MTR	0.9728
	LT	0.9794

7 CONCLUSION

One of the novel methods to improve the surface quality of materials is electro-discharge coating. In this process, the material removal rate as an index of the processing speed, and the average coated layer thickness as an index of the surface quality are important parameters and tuning the input parameters, which are pulse-on time, pulse-off time, current intensity, concentration pressure, and sintering temperature, so as attaining the desired value of them is an important task in this process. Two independent models are established for the two outputs and therefore, the models do not have any relation with each other. As a result of the nonlinearity of the effect of the input parameters on the outputs and the wide numerical range of them, the development of a precise predictive model of the outputs based on the input parameters can be beneficial. In this research, a comparative study of three powerful machine learning algorithms, RVM, ELM and LS-SVM for this purpose has been investigated. Error analysis of the three models suggested that the highest degree of accuracy can be obtained by the LS-SVM models, even in comparison with the previous ANN-based models. The values R^2 above 0.99 for the training data and above 0.97 for the test data show the high accuracy and generalization capability degree related to the LS-SVM models, which can be applied for the input parameters tuning in order to attain a preferred value of the outputs.

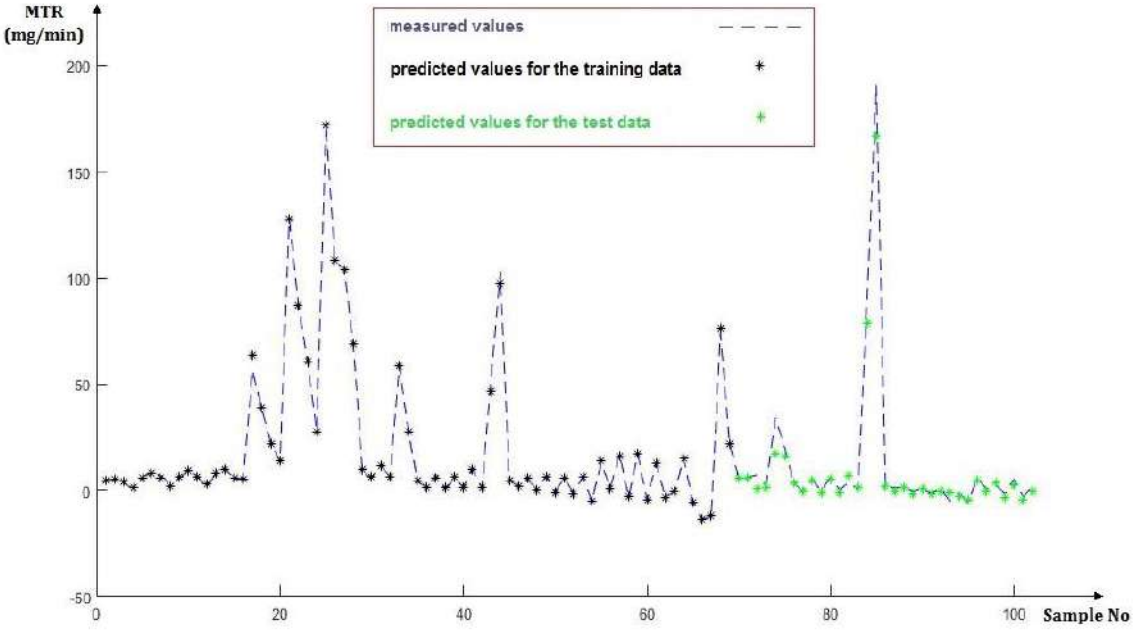

Fig. 7. The predicted values of MTR alongside with the corresponding targets.

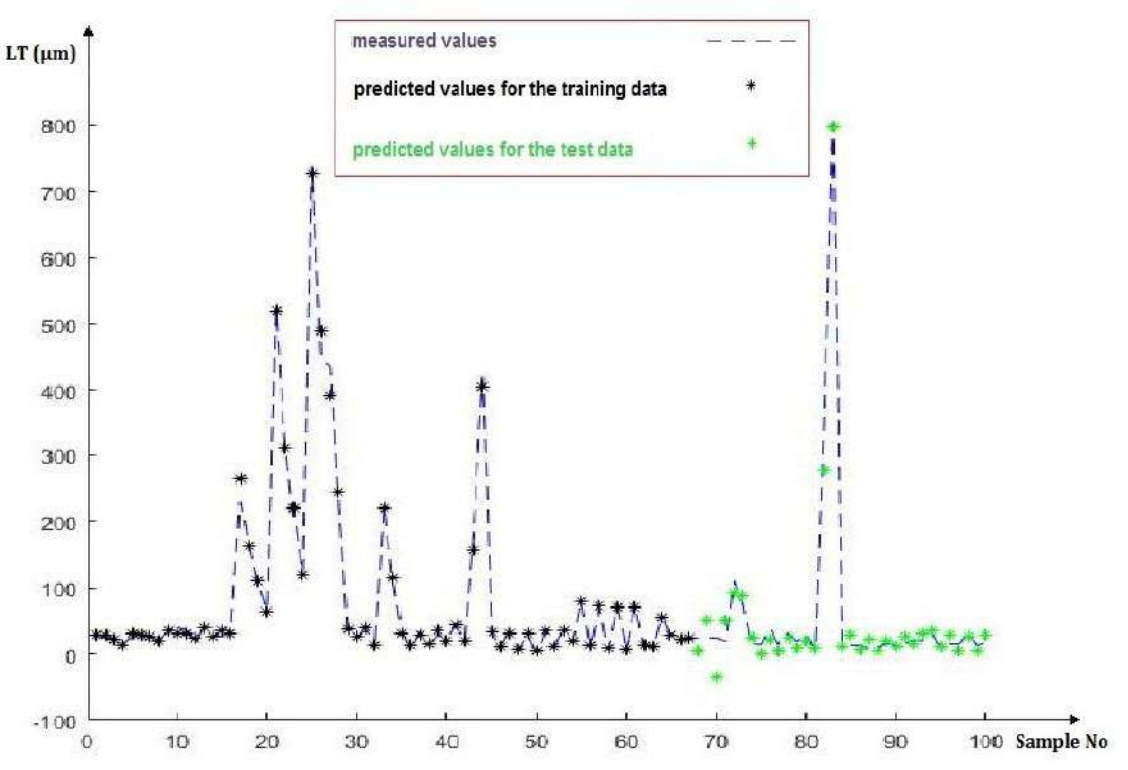

Fig. 8. The predicted values of LT alongside with the corresponding targets.

REFERENCES

[1] Jameson, E. C., Electrical Discharge Machining: Society of Manufacturing Engineers, 2001.

[2] Ho, K., Newman, S., State of the Art Electrical Discharge Machining (EDM), International Journal of Machine Tools and Manufacture, Vol. 43, No. 13, 2003, pp. 1287-1300.

[3] Kumar, S., Singh, R., Singh, T., and Sethi, B., Surface Modification by Electrical Discharge Machining: A Review, Journal of Materials Processing Technology, Vol. 209, No. 8, 2009, pp. 3675-3687.

[4] Janmanee, P., Muttamara, A., Surface Modification Of Tungsten Carbide by Electrical Discharge Coating (EDC) Using a Titanium Powder Suspension, Applied Surface Science, Vol. 258, No. 19, 2012, pp. 7255-7265.

[5] Beri, N., Maheshwari, S., Sharma, C., and Kumar, A., Technological Advancement in Electrical Discharge Machining with Powder Metallurgy Processed Electrodes: A Review, Materials and Manufacturing Processes, Vol. 25, No. 10, 2010, pp. 1186-1197.

[6] Das, A.,Misra, J. P., Experimental investigation On Surface Modification of Aluminum by Electric Discharge Coating Process Using TiC/Cu Green Compact Tool-Electrode, Machining Science and Technology, Vol. 16, No. 4, 2012, pp. 601-623.

[7] Naik, S. K., Study the Effect of Compact Pressure of P/M Tool Electrode on Electro Discharge Coating (EDC) Process, 2013.

[8] Hwang, Y. L., Kuo, C. L., and Hwang, S. F., The Coating of Tic Layer On the Surface of Nickel by Electric Discharge Coating (EDC) with a Multi-Layer Electrode, Journal of Materials Processing Technology, Vol. 210, No. 4, 2010, pp. 642-652.

[9] Patowari, P. K., Saha, P., and Mishra, P., Artificial Neural Network Model in Surface Modification by EDM Using Tungsten–Copper Powder Metallurgy Sintered Electrodes, The International Journal of Advanced Manufacturing Technology, Vol. 51, No. 5, 2010, pp. 627-638.

[10] Provost, F., Kohavi, R., Glossary of Terms, Journal of Machine Learning, Vol. 30, No. 2-3, 1998, pp. 271-274.

[11] Hastie, T., Tibshirani, R., and Friedman, J., Overview of supervised Learning, The Elements of Statistical Learning, 2009, pp. 9-41.

[12] Tyagi, R.,Kumar, S., Kumar, V., Mohanty, S., Das, A.,and Mandal, A., Analysis and Prediction of Electrical Discharge Coating Using Artificial Neural Network (ANN), Advances in Simulation, Product Design and Development, 2020, pp. 177-189.

[13] Sahu, A. K., Mahapatra, S. S., and Chatterjee, S., Optimization of Electro-Discharge Coating Process Using Harmony Search, Materials Today: Proceedings, Vol. 5, No. 5, 2018, pp. 12673-12680.

[14] Vapnik, V., The Nature of Statistical Learning Theory: Springer Science & Business Media, 2013.

[15] Bishop, C. M., Pattern Recognition And Machine Learning: Springer, 2006.

[16] Meyer, D., Leisch, F., and Hornik, K., The Support Vector Machine Under Test, Neurocomputing, Vol. 55, No. 1, 2003, pp. 169-186.

[17] Cristianini, N., Shawe-Taylor, J., An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods: Cambridge University Press, 2000.

[18] Parrella, F., Online Support Vector Regression, Master's Thesis, Department of Information Science, University of Genoa, Italy, 2007.

[19] Kecman, V., Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models: MIT Press, 2001.

[20] Suykens, J. A., Van Gestel, T., and De Brabanter, J., Least Squares Support Vector Machines: World Scientific, 2002.

[21] Wang, H., Hu, D., Comparison of SVM and LS-SVM for Regression, Vol. 1, 2005, pp. 279-283.

[22] Ding, S., Zhao, H., Zhang, Y., Xu, X., and Nie, R., Extreme Learning Machine: Algorithm, Theory and Applications, Artificial Intelligence Review, Vol. 44, No. 1, 2015, pp. 103-115.

[23] Huang, G. B., Zhu, Q. Y., and Siew, C. K., Extreme Learning Machine: Theory and Applications, Neurocomputing, Vol. 70, No. 1-3, pp. 489-501, 2006.

[24] Sun, Z. L., Choi, T. M., Au, K .F., and Yu, Y., Sales Forecasting Using Extreme Learning Machine with Applications in Fashion Retailing, Decision Support Systems, Vol. 46, No. 1, 2008, pp. 411-419.

[25] Huang, G. B., Learning Capability and Storage Capacity of Two-Hidden-Layer Feedforward Networks, IEEE Transactions on Neural Networks, Vol. 14, No. 2, 2003, pp. 274-281.

[26] Liang, N. Y., Huang, G. B., Saratchandran, P., and Sundararajan, N., A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks, IEEE Transactions on Neural Networks, Vol. 17, No. 6, 2006, pp. 1411-1423.

[27] Wan, C., Xu, Z., Pinson, P., Dong, Z. Y., and Wong, K. P., Probabilistic Forecasting of Wind Power Generation Using Extreme Learning Machine, IEEE Transactions on Power Systems, Vol. 29, No. 3, 2013, pp. 1033-1044.

[28] Caesarendra, W., Widodo, A., and Yang, B. S., Application of Relevance Vector Machine and Logistic Regression for Machine Degradation Assessment, Mechanical Systems and Signal Processing, Vol. 24, No. 4, 2010, pp. 1161-1171.

[29] Widodo, A., Kim, E. Y., Son, J. D., Yang, B. S., Tan, A. C., Gu, D. S., Choi, B. K., and Mathew, J., Fault Diagnosis of Low Speed Bearing Based On Relevance Vector Machine And Support Vector Machine, Expert systems with Applications,Vol. 36, No. 3, 2009, pp. 7252-7261.

[30] Wang, X., Ye, M., and Duanmu, C., Classification of Data From Electronic Nose Using Relevance Vector Machines, Sensors and Actuators B: Chemical, Vol. 140, No. 1, 2009, pp. 143-148.

[31] Jiang, J., Li, M., Jing, X., and Lv, B., Research on the Performance of Relevance Vector Machine for Regression and Classification, 2015, pp. 758-762.

[32] MacKay, D. J., Bayesian Interpolation, Neural Computation, Vol. 4, No. 3, 1992, pp. 415-447.

[33] Thayananthan, A., Relevance Vector Machine Based Mixture of Experts, Cambridge: Department of Engineering, University of Cambridge, 2005.

[34] Tipping, M. E., Faul, A. C., Fast Marginal Likelihood Maximisation For Sparse Bayesian Models, In C. M. Bishopand B. J. Frey (Eds.), Proceedings of the Ninth International Workshop on ArtificialIntelligence and Statistics, Key West, FL, Jan 3–6, 2003.

[35] Tipping, M. E., SPARSEBAYES V1. 1: A Baseline Matlab Implementation of Sparse Bayesian Model Estimation, 2009.

[36] Pelckmans, K., Suykens, J. A., Van Gestel, T., De Brabanter, J., Lukas, L., Hamers, B., De, B., Moor, and Vandewalle, J., LS-SVMlab: a Matlab/C Toolbox for Least Squares Support Vector Machines, Tutorial. KULeuven-ESAT. Leuven, Belgium, Vol. 142, 2002, pp. 1-2.