Thailand Statistician
October 2020; 18(4): 381-391
http://statassoc.or.th
Contributed paper

Modelling and Analysis of Simple Pendulum Computer Experiments Using a Support Vector Regression Model

Kazeem Adewale Osuolale
Nigerian Institute of Medical Research, Lagos, Nigeria.
*Corresponding author; e-mail: whereisqosimadewale@gmail.com

Received: 20 May 2019
Revised: 24 June 2019
Accepted: 11 July 2019

Abstract

In many practical situations, we know the dynamics of the analyzed system. Based on this known dynamics, we can build a computer-based model simulating the system’s behaviour. In principle, it is possible to directly use this model to predict the system’s behaviour, but in many cases, the model is very computationally complicated, and its direct use would require a large amount of computation time on high performance computers. A natural way to drastically reduce the computation time needed for the prediction of the system’s behaviour is to use machine learning to come up with a simpler faster-to-compute model. To train the corresponding machine learning algorithm, we can use the results of applying the existing (complex) computer-based model to different inputs−and these inputs can be selected by an appropriate experimental design technique. In this paper, we illustrate this methodology on a simple example of a pendulum; as machine learning techniques, we use support vector machine method. The results show that this methodology indeed drastically reduces the computation time and still provides reasonably accurate results.

Keywords: Computation time, Gaussian radial basis function, machine learning, orthogonal array-based Latin hypercube design, simple pendulum model.

1. Introduction

In many practical situations, we know the dynamics of the analyzed system. Based on this known dynamics, we can build a computer-based model simulating the system’s behaviour. In principle, it is possible to directly use this model to predict the system's behaviour, but in many cases, the model is very computationally complicated, and its direct use would require a large amount of computation time on high performance computers. A natural way to drastically reduce the computation time needed for the prediction of the system's behaviour is to use machine learning to come up with a simpler faster-to-compute model. To train the corresponding machine learning algorithm, we can use the results of applying the existing (complex) computer-based model to different inputs−and these inputs can be selected by an appropriate experimental design technique. In this paper, we illustrate this methodology on a simple example of a pendulum; as machine learning techniques, we use support vector machine method. The results show that this methodology indeed drastically reduces the computation time and still provides reasonably accurate results.

Conventional experimental design has its origin from the theory of design of experiments (DOE) when physical experiments are conducted (Montgomery 2001) while space-filling designs are connected with computer experiments. Computer experiments differ greatly from physical experiments because they deal with functions that are considered to have more complex behaviour with no random error since the typical computer experiments are deterministic in nature. For instance, the simple pendulum model considered in this study is deterministic.

Computer experiments are more commonly employed in engineering, science and technology because the conventional physical experiments could require more time, money and some other resources for its implementation. In some instances, computer experiments may be performed prior to physical experiments so that computer experiment can serve as a prototype before embarking on physical experiments. A computer experiment is carried out by using data obtained from a mathematical model, known as a computer model in place of the physical process. Osuolale et al. (2014a) quoted Strogatz (2003) to have reported that the first computer experiment was performed by Enrico Fermi and colleagues at the Los Alamos Scientific Laboratory in 1953. In this present study, a simple pendulum computer experiment that utilizes a mathematical model that is used to mimic the simple pendulum experiment that is usually performed in the laboratory and a support vector regression (SVR) model is thereafter adopted to emulate the simple pendulum computer model in order to save time that may be required by a computer code. Predictions of the stoppage time of pendulum at untried inputs were also made using SVR model. Building this metamodel is one of the purposes of computer experiments. Usually, a pendulum is suspended by a massless string from some point about which it is allowed to swing back and forth in a place when performed in the laboratory. The orthogonal array Latin hypercube design (OALHD) originally constructed by Osuolale et al. (2015b) is used to develop a computer experiment and the SVR model is used as a metamodel to emulate the simple pendulum computer model. The OA (49, 3) LHD was adopted to enhance the performance of the pendulum model.

2. Material and Methods

The model development and analysis in this study were performed using MATLAB 2003 software (MATLAB Basics 2003). Orthogonal array-based Latin hypercube design (OALHD) was used to develop a computer experiment using a simple pendulum model and the SVR model was subsequently used to emulate a simple pendulum computer model. The results of the orthogonal array-based Latin hypercube design, OA (49, 3) LHD and its plot for projection properties among the three input variables as constructed by Osuolale et al. (2014b) are provided in Table 1 and Figure 1, respectively. A simple pendulum experiment has been proposed as a novel application in the realm of computer experiments by Osuolale et al. (2014a) and Osuolale et al. (2015a) have also investigated a simple pendulum model using a Gaussian stochastic model as a metamodel. Recently, Osuolale (2018) employed a support vector regression model as a computer based metamodel to mimic the simple pendulum computer model in order to reduce the required computational efforts in running the computer codes and predict the stoppage time of pendulum at untried input values.

Table 1 OA (49, 3) LHD constructed for simple pendulum computer experiment

OA (49, 3) LHD							
Runs	X_1	X_2	X_3	Runs	X_1	X_2	X_3
1	0.0102	0.0102	0.0102	26	0.5204	0.6429	0.0714
2	0.0306	0.1531	0.1531	27	0.5408	0.7857	0.2143
3	0.0510	0.2959	0.2959	28	0.5612	0.9286	0.3571
4	0.0714	0.4388	0.4388	29	0.5816	0.0918	0.6633
5	0.0918	0.5816	0.5816	30	0.6020	0.2347	0.8061
6	0.1122	0.7245	0.7245	31	0.6224	0.3776	0.9490
7	0.1327	0.8673	0.8673	32	0.6429	0.5204	0.0918
8	0.1531	0.0306	0.1735	33	0.6633	0.6633	0.2347
9	0.1735	0.1735	0.3163	34	0.6837	0.8061	0.3776
10	0.1939	0.3163	0.4592	35	0.7041	0.9490	0.5204
11	0.2143	0.4592	0.6020	36	0.7245	0.1122	0.8265
12	0.2347	0.6020	0.7449	37	0.7449	0.2551	0.9694
13	0.2551	0.7449	0.8878	38	0.7653	0.3980	0.1122
14	0.2755	0.8878	0.0306	39	0.7857	0.5408	0.2551
15	0.2959	0.0510	0.3367	40	0.8061	0.6837	0.3980
16	0.3163	0.1939	0.4796	41	0.8265	0.8265	0.5408
17	0.3367	0.3367	0.6224	42	0.8469	0.9694	0.6837
18	0.3571	0.4796	0.7653	43	0.8673	0.1327	0.9898
19	0.3776	0.6224	0.9082	44	0.8878	0.2755	0.1327
20	0.3980	0.7653	0.0510	45	0.9082	0.4184	0.2755
21	0.4184	0.9082	0.1939	46	0.9286	0.5612	0.4184
22	0.4388	0.0714	0.5000	47	0.9490	0.7041	0.5612
23	0.4592	0.2143	0.6429	48	0.9694	0.8469	0.7041
24	0.4796	0.3571	0.7857	49	0.9898	0.9898	0.8469
25	0.5000	0.5000	0.9286				

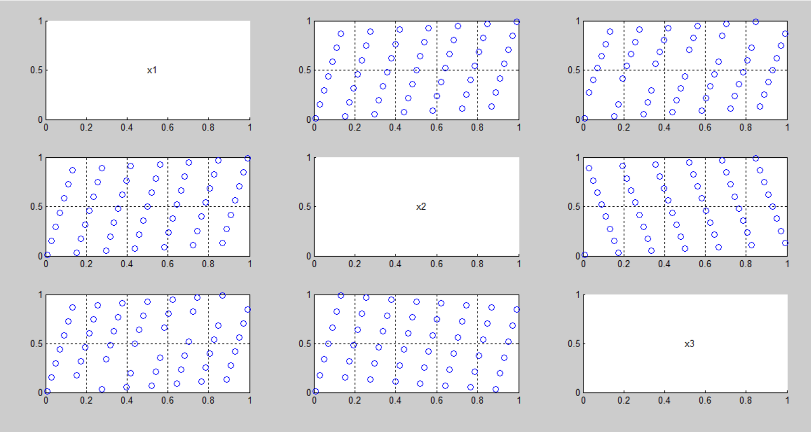

Figure 1 Projection properties of OA (49, 3) LHD

The OA (49, 3) LHD was scaled according to the assumed range for the input variables as shown in Table 2 using:

$$y_{OALHD} = \frac{y_{data} - y_{data(\min)}}{y_{data(\max)} - y_{data(\min)}}, \tag{1}$$

$$y_{data} = y_{OALHD}\left(y_{data(\max)} - y_{data(\min)}\right) + y_{data(\min)}.$$

Table 2 Input variables for simple pendulum model

Variable	Variable name	Minimum	Maximum
$X_1(L)$	Length (m)	0.20	0.49
$X_2(\theta)$	Angle (degree)	5.00	42.59
$X_3(M)$	Mass (kg)	0.01	0.0459
Y	Stoppage Time (s)	-	-

The scaled OALHD was used to develop a simple pendulum computer experiment using the non-linear differential equation as its model to produce the output of a computer experiment.

$$ML^2\frac{d^2\theta}{dt^2} + b\frac{d\theta}{dt} + MgL\sin(\theta) = 0, \quad \frac{d^2\theta}{dt^2} = \frac{-b\frac{d\theta}{dt} - MgL\sin(\theta)}{ML^2}, \tag{2}$$

where $y = \frac{d^2\theta}{dt^2}$ is stoppage time of the pendulum bob,

M (kg) is mass of the pendulum bob,
L (m) is length of the pendulum,
θ (degree) is the pendulum displacement angle,
b is coefficient of friction (Ns/rad),
g is acceleration (ms^{-2}) due to gravity.

The coefficient of friction (b) is fixed at 0.02 Ns/rad and the acceleration due to gravity (g) is $9.81\ \text{ms}^{-2}$. The scaled input variables and the output simulated from a computer model now constitute the experimental results for the training data sets.

3. Modelling Simple Pendulum Computer Experiments

A computer model is a model of a physical process which requires a metamodel to lessen the computational efforts in using computer codes. A metamodel is simply a model adopted to emulate another model. A wide variety of techniques, as stated by Osuolale (2018), has been discussed in the literature for creating the metamodels. These techniques include response surface modelling, e.g., Osuna et al. (1997), radial basis functions, Dyn et al. (1986) and Fang and Horstemeyer (2006) and multivariate adaptive regression splines, Friedman (1991). In this study, a support vector regression (SVR) model was investigated as an alternative technique for approximating a simple pendulum computer model. The SVR technique is an extension of the famous classification tool used in support vector machines (SVM). A SVM is a machine learning tool that has its origin in statistical learning theory (Cortes and Vapnik 1995). The development of SVMs was originally meant to solve the classification problem, but they have also been used to deal with regression problems (Vapnik et al. 1997). Several researchers have applied it to solve classification problems while previous researchers applied it to optical character recognition and face detection (Scholkopf et al. 1996, Osuna et al. 1997). The SVR model works by performing a non-linear mapping of the data from the input space to a higher dimensional feature space where linear regression can be performed. The choice of kernel function for a support vector regression is conventionally by trial and error as this may involve searching for the best parameters for the kernels through grid search and applying each of the kernels for regression. The SVR algorithm written by Gunn (1998) was adopted for modelling and analysing simple pendulum computer experiments with Gaussian Radial Basis Function (GRBF) used as a kernel function to ensure that the SVR efficiently captures non-linearities in the simple pendulum computer model. A SVR simply constructs a hyperplane that passes near each design point such that they fall within a specified distance of the hyperplane. The hyperplane is then used to predict other responses. A SVR estimates the real function as

$$y = r(x) + \psi, \tag{3}$$

where ψ is an independent random noise, x is the multivariate input, y is the scalar output and r is the regression function.

Table 3 Experimental data for simple pendulum computer experiment (training datasets)

Run	X_1	X_2	X_3	Y	Run	X_1	X_2	X_3	Y
1	0.2270	8.4470	0.0119	8.8500	26	0.3499	27.8905	0.0123	5.4000
2	0.2319	12.8375	0.0129	8.2200	27	0.3548	32.2809	0.0133	4.6500
3	0.2368	17.2279	0.0139	7.5800	28	0.3597	36.6714	0.0144	3.8000
4	0.2418	21.6184	0.0150	6.9700	29	0.3647	10.9558	0.0166	2.3900
5	0.2467	26.0088	0.0160	6.3900	30	0.3696	15.3463	0.0176	2.9600
6	0.2516	30.3993	0.0170	5.8500	31	0.3745	19.7368	0.0186	3.5100
7	0.2565	34.7898	0.0181	5.3300	32	0.3794	24.1272	0.0125	4.4200
8	0.2614	9.0742	0.0130	6.9100	33	0.3843	28.5177	0.0135	3.3100
9	0.2663	13.4647	0.0141	6.4000	34	0.3892	32.9081	0.0145	2.8400
10	0.2713	17.8551	0.0151	5.8700	35	0.3942	37.2986	0.0155	3.2500
11	0.2762	22.2456	0.0161	5.3400	36	0.3991	11.5830	0.0178	2.6300
12	0.2811	26.6360	0.0172	4.8200	37	0.4040	15.9735	0.0188	3.3100
13	0.2860	31.0265	0.0182	4.2900	38	0.4089	20.3640	0.0126	2.4800
14	0.2909	35.4170	0.0120	7.3900	39	0.4138	24.7544	0.0136	2.6200
15	0.2958	9.7014	0.0142	5.3500	40	0.4187	29.1449	0.0147	3.1600
16	0.3008	14.0919	0.0153	4.8700	41	0.4237	33.5353	0.0157	3.7300
17	0.3057	18.4823	0.0163	4.3500	42	0.4286	37.9258	0.0167	3.5000
18	0.3106	22.8728	0.0173	3.7500	43	0.4335	12.2102	0.0189	2.4000
19	0.3155	27.2633	0.0183	2.6800	44	0.4384	16.6007	0.0128	2.5400
20	0.3204	31.6537	0.0122	6.3600	45	0.4433	20.9912	0.0138	3.1300
21	0.3253	36.0442	0.0132	5.6300	46	0.4482	25.3816	0.0148	3.7300
22	0.3303	10.3286	0.0154	3.8600	47	0.4532	29.7721	0.0158	4.3400
23	0.3352	14.7191	0.0164	3.1000	48	0.4581	34.1626	0.0169	4.1900
24	0.3401	19.1095	0.0175	2.7800	49	0.4630	38.5530	0.0179	4.8600
25	0.3450	23.5000	0.0185	3.1200					

The SVR technique selects the best approximate model from a group of selection models that minimize the prediction risk. Linear and nonlinear regression can be performed in SVR. When a linear regression is employed, the pool of approximation models is given by

$$\hat{f}(x) = (w.x) + b, \tag{4}$$

where b is the bias term and $(w.x)$ is the dot product of two points; that is, w and $x.$ By minimizing the empirical risk using the ε-insensitive loss function will allow regression estimates. It is desirable to have a flat approximation function and this can be achieved by minimizing $\|w\|^2$. Non-negative slack variables α_i and α_i^* are also introduced to account for training points that fall outside of the ε -insensitive zone. Then the SVR problem can be formulated as the minimization of the following function

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\left(\xi_i + \xi_i^*\right), \tag{5}$$

such that

$$\begin{cases} y_i - (w.x_i) - b \le \varepsilon + \xi_i^*, \\ (w.x_i) + b - y_i \le \varepsilon + \xi_i, \\ \xi_i^*, \xi_i \ge 0, \end{cases} \tag{6}$$

where C is a positive constant and ε is the insensitive zone chosen by the user. C can also be referred to as the penalty parameter. The parameter ε determines the width of the ε -insensitive zone and affects the complexity of the model. The optimization problem in (5) and the constraints in (6) can be written as a Lagrangian function

$$\begin{aligned} L = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\left(\xi_i + \xi_i^*\right) - \sum_{i=1}^{n}\alpha_i(\varepsilon + \xi_i - y_i + (w.x_i) + b) \\ - \sum_{i=1}^{n}\alpha_i^*(\varepsilon + \xi^*_i + y_i - (w.x_i) - b) - \sum_{i=1}^{n}\left(v_i\xi_i + v_i^*\xi_i^*\right), \end{aligned} \tag{7}$$

where v_i and v_i^* are additional slack variables. The dual form optimization problem is given as

$$\text{maximize}\left\{-\frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)(x_i.x_j) - \varepsilon\sum_{i=1}^{n}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{n}y_i(\alpha_i - \alpha_i^*)\right\}$$

such that

$$\begin{cases} \sum_{i=1}^{n}(\alpha_i - \alpha_i^*) = 0, \\ (\alpha_i - \alpha_i^*) \in [0, C], \end{cases} \tag{8}$$

and w is given as

$$w = \sum_{i=1}^{n}(\alpha_i^* - \alpha_i)x_i. \tag{9}$$

The linear regression expressed in (4) is written as

$$f(x) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)(x_i.x_i) + b, \tag{10}$$

where $(x_i.x_i)$ is the dot products of two points and the variables α_i, α_i^* and b are calculated by the SVR algorithm (Smola and Scholkopf 2004). A nonlinear regression model can also be developed by replacing the dot product $(x_i.x_i)$ with a kernel function K. The optimization problem in (8) can be rewritten as

$$\text{maximize}\left\{-\frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(x_i.x_j) - \varepsilon\sum_{i=1}^{n}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{n}y_i(\alpha_i - \alpha_i^*)\right\}$$

such that

$$\begin{cases} \sum_{i=1}^{n}(\alpha_i - \alpha_i^*) = 0, \\ (\alpha_i - \alpha_i^*) \in [0, C]. \end{cases} \tag{11}$$

Replacing the dot product $(x_i.x_i)$ with a kernel function, K, in the approximation function, (10) gives the nonlinear SVR approximation as

$$f(x) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)K(x_i.x_j) + b, \tag{12}$$

where $K(x_i.x_j) = \exp\left(-\frac{\| x_i - x_j^T \|^2}{2p^2}\right)$ is the Gaussian radial basis function. The algorithm adopted implements the model as

$$\{N_{sv,}\beta, bias\} = f(X_{data}(X), Y_{data}(Y), \text{kernel}(Ker), \text{Cost}(C), \text{loss}, \text{insensitivity}(\varepsilon)). \tag{13}$$

Accuracies of SVR model were checked using relative average absolute error (RAAE) and relative maximum absolute error (RMAE), respectively as given below

(i) RAAE

$$\text{RAAE} = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n.SD}, \tag{14}$$

(ii) RMAE

$$\text{RMAE} = \frac{\max\left(|y_1 - \hat{y}_1|, |y_2 - \hat{y}_2|, \ldots, |y_n - \hat{y}_n|\right)}{SD}. \tag{15}$$

4. Analysis of Simple Pendulum Computer Experiment

It is clear in Table 3 that the three input variables involved in the simple pendulum computer experiment did not have the same scales. These variables were normalized to lessen the dimension effect of each design variable and prevent an inconsistent prediction of the SVR model. The normalized experimental data based on the simple pendulum computer experiment are given in Tables 4 and 5, respectively.

The SVR model using the Gaussian RBF trained with 77.60% of the experimental runs as shown in Table 5. The Gaussian RBF has zero bias. The SVR with Gaussian RBF kernel function indicated that not all the data generated from the experimental runs are support vectors. The estimated values of RAAE (0.0604) and RMAE (0.2686) provided by the GRB function were small which shows some levels of accuracy of SVR model. The regression model was controlled by three methods, namely, the loss function, the kernel as earlier discussed and additional capacity control, C. The e-insensitive loss function was used with $e = 0.05$ and $C = 8$. A simple pendulum computer experiment was performed with a new set of data to assess the prediction accuracy of the SVR model. This assessment also enables us to find out how well the SVR model predicts the output at untried inputs. The assumed range of design variables for the test data was given in Table 6 followed by the normalized experimental data for simple pendulum computer experiment in Table 7, respectively.

Table 4 Normalized experimental data for simple pendulum computer experiment (training datasets)

Run	X_1	X_2	X_3	Y	Run	X_1	X_2	X_3	Y
1	−1.6797	−1.6797	−1.6797	2.6254	26	0.0700	0.4899	−1.4697	0.5557
2	−1.6097	−1.1898	−1.1898	2.2474	27	0.1400	0.9798	−0.9798	0.1058
3	−1.5397	−0.6999	-0.6999	1.8635	28	0.2100	1.4697	−0.4899	−0.4041
4	−1.4697	−0.2100	−0.2100	1.4975	29	0.2799	−1.3997	0.5599	−1.2500
5	−1.3997	0.2799	0.2799	1.1496	30	0.3499	−0.9098	1.0498	−0.9081
6	−1.3297	0.7698	0.7698	0.8257	31	0.4199	−0.4199	1.5397	−0.5781
7	−1.2597	1.2597	1.2597	0.5137	32	0.4899	0.0700	−1.3997	−0.0322
8	−1.1898	−1.6097	−1.1198	1.4615	33	0.5599	0.5599	−0.9098	−0.6981
9	−1.1198	−1.1198	−0.6299	1.1556	34	0.6299	1.0498	−0.4199	−0.9800
10	−1.0498	−0.6299	−0.1400	0.8377	35	0.6999	1.5397	0.0700	−0.7341
11	−0.9798	−0.1400	0.3499	0.5197	36	0.7698	−1.3297	1.1198	−1.1060
12	−0.9098	0.3499	0.8398	0.2078	37	0.8398	−0.8398	1.6097	−0.6981
13	−0.8398	0.8398	1.3297	−0.1102	38	0.9098	−0.3499	−1.3297	−1.1960
14	−0.7698	1.3297	−1.6097	1.7495	39	0.9798	0.1400	−0.8398	−1.1120
15	−0.6999	−1.5397	−0.5599	0.5257	40	1.0498	0.6299	−0.3499	−0.7881
16	−0.6299	−1.0498	−0.0700	0.2378	41	1.1198	1.1198	0.1400	−0.4461
17	−0.5599	−0.5599	0.4199	−0.0742	42	1.1898	1.6097	0.6299	−0.5841
18	−0.4899	−0.0700	0.9098	−0.4341	43	1.2597	−1.2597	1.6797	−1.2440
19	−0.4199	0.4199	1.3997	−1.0760	44	1.3297	−0.7698	−1.2597	−1.1600
20	−0.3499	0.9098	−1.5397	1.1316	45	1.3997	−0.2799	−0.7698	−0.8061
21	−0.2799	1.3997	−1.0498	0.6937	46	1.4697	0.2100	−0.2799	−0.4461
22	−0.2100	−1.4697	0.0000	−0.3681	47	1.5397	0.6999	0.2100	−0.0802
23	−0.1400	−0.9798	0.4899	−0.8241	48	1.6097	1.1898	0.6999	−0.1702
24	−0.0700	−0.4899	0.9798	−1.0160	49	1.6797	1.6797	1.1898	0.2318
25	0.0000	0.0000	1.4697	−0.8121					

Table 5 Results of SVR for the simple pendulum computer experiment

SVR Using GRBF			
SV	38 (77.60%)	C	8.00
Bias	0.000	e	0.05
Sum Beta	2.8084		
$\|w_0\|^2$	28.1744		

Table 6 Input and output variables for simple pendulum model (test)

Variable	Variable Name	Minimum	Maximum
$X_1(L)$	Length (m)	0.5	1.0
$X_2(\theta)$	Angle (degree)	43.6	90.0
$X_3(M)$	Mass (kg)	0.049	0.100
Y	Stoppage Time (s)	-	-

Based on the normalized data presented in Table 7, the predicted stoppage time of pendulum, y is calculated and the fitted SVR model gave predicted values that were close to the simulated test data. The graph in Figure 2 shows the plot of the predicted versus simulated output.

Table 7 Normalized experimental data for simple pendulum experiment (test)

Run	X_1	X_2	X_3	Y	Run	X_1	X_2	X_3	Y
1	−1.6797	−1.6797	−1.6797	−1.5338	26	0.0700	0.4899	−1.4697	−0.5247
2	−1.6097	−1.1898	−1.1898	−1.2452	27	0.1400	0.9798	−0.9798	−0.1181
3	−1.5397	−0.6999	−0.6999	−1.8506	28	0.2100	1.4697	−0.4899	−0.3629
4	−1.4697	−0.2100	−0.2100	−1.0515	29	0.2799	−1.3997	0.5599	0.4295
5	−1.3997	0.2799	0.2799	−0.9810	30	0.3499	−0.9098	1.0498	0.5817
6	−1.3297	0.7698	0.7698	−1.0240	31	0.4199	−0.4199	1.5397	0.8020
7	−1.2597	1.2597	1.2597	−1.0099	32	0.4899	0.0700	−1.3997	0.4206
8	−1.1898	−1.6097	−1.1198	−1.4106	33	0.5599	0.5599	−0.9098	0.5075
9	−1.1198	−1.1198	−0.6299	−1.0493	34	0.6299	1.0498	−0.4199	0.7983
10	−1.0498	−0.6299	−0.1400	−1.0953	35	0.6999	1.5397	0.0700	-0.6085
11	−0.9798	−0.1400	0.3499	−0.6085	36	0.7698	−1.3297	1.1198	0.0036
12	−0.9098	0.3499	0.8398	−0.5328	37	0.8398	−0.8398	1.6097	0.9823
13	−0.8398	0.8398	1.3297	−0.8163	38	0.9098	−0.3499	−1.3297	0.6573
14	−0.7698	1.3297	−1.6097	−0.6196	39	0.9798	0.1400	−0.8398	1.0936
15	−0.6999	−1.5397	−0.5599	−0.3718	40	1.0498	0.6299	−0.3499	0.7672
16	−0.6299	−1.0498	−0.0700	−0.7836	41	1.1198	1.1198	0.1400	0.2277
17	−0.5599	−0.5599	0.4199	−0.0921	42	1.1898	1.6097	0.6299	0.5282
18	−0.4899	−0.0700	0.9098	−0.5076	43	1.2597	−1.2597	1.6797	1.8705
19	−0.4199	0.4199	1.3997	0.1936	44	1.3297	−0.7698	−1.2597	0.8280
20	−0.3499	0.9098	−1.5397	−0.4757	45	1.3997	−0.2799	−0.7698	0.9193
21	−0.2799	1.3997	−1.0498	0.0452	46	1.4697	0.2100	−0.2799	0.7249
22	−0.2100	−1.4697	0.0000	−0.4564	47	1.5397	0.6999	0.2100	2.0419
23	−0.1400	−0.9798	0.4899	0.2604	48	1.6097	1.1898	0.6999	2.3580
24	−0.0700	−0.4899	0.9798	0.0875	49	1.6797	1.6797	1.1898	2.6044
25	0.0000	0.0000	1.4697	−0.6033					

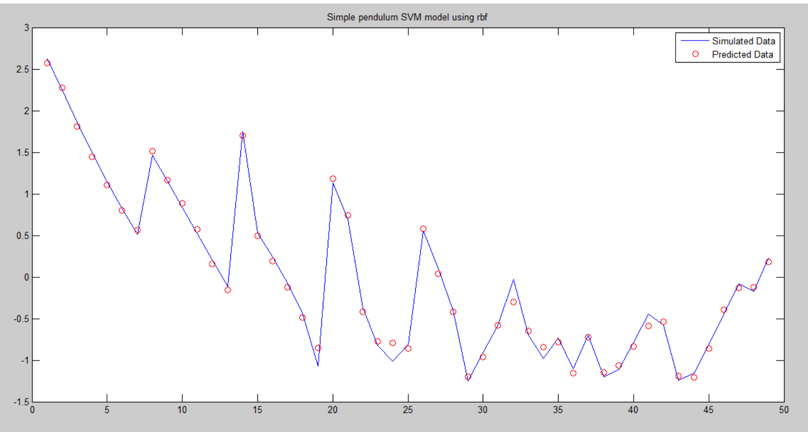

Figure 2 Graph of the predicted y against simulated output using Gaussian RBF K.F.

5. Discussion

The constructed OA (49, 3) LHD were used to develop a simple pendulum computer experiment by scaling OA (49, 3) LHD and then applied equation (1) to obtain the experimental output. The scaled input variables and the output simulated from a computer model now constitute the experimental results for the training data sets. The Support Vector Regression (SVR) model was fitted using the training data sets. The SVR model showed that that GRBF trained with 77.60% of the experimental runs with zero bias and the estimated values of RAAE (0.0604) and RMAE (0.2686) were small confirming some levels of accuracy of SVR model. The fitted SVR model gave predicted values that were close to the test data. The test data was also simulated using a 49-run experimental design in order to test the accuracy of the SVR model. These test data were used on the fitted SVR model to predict the stoppage time of pendulum and the result of the predictions were close to the test data as shown in Figure 2. This shows that the fitted SVR model is fairly efficient at modelling and predicting the stoppage time of a pendulum.

6. Conclusions

This study presents modelling and analysis of simple pendulum computer experiments using a support vector regression model. The OA (49, 3) LHD was implemented using a support vector regression (SVR) model. The SVR model with Gaussian RBF kernel function modelled the simple pendulum computer experimental data fairly well with zero bias and predicted the stoppage time of simple pendulum fairly well though the predicted output did not give exact estimates for the simulated output. The SVR model predicted the simulated output differently and is therefore regarded as an approximate model (metamodel) of a simple pendulum model. This study concluded that SVR model with Gaussian radial basis function is good at modelling and predicting the stoppage time of simple pendulum using a 49-run experimental design. The results obtained in this study show that this methodology indeed drastically reduces the computation time and still provides reasonably accurate results. Applications of the Support Vector Regression (SVR) model with kernel functions different from the Gaussian Radial Basis Function (GRBF) so as to compare the performance of SVR model and predict the output of a simple pendulum computer experiment at untried inputs will be considered for future research.

Acknowledgements

My profound gratitude goes to my colleague, Kunle Ifeta for supporting me in the development and implementation of the algorithm in MATLAB programming language. Many thanks to Professors B.L. Adeleke and W.B. Yahya of the University of Ilorin for their useful suggestions.

References

Cortes C, Vapnik V. Support vector networks. Mach Learn. 1995; 20: 273-297.

Dyn N, Levin D, Rippa S. Numerical procedures for surface fitting of scattered data by radial basis functions. SIAM J Sci Comput. 1986; 7: 639-659.

Fang H, Horstemeyer MF. Global response approximation with radial basis functions. Eng Optimiz. 2006; 38: 407-424.

Friedman JH. Multivariate adaptive regressive splines. Ann Stat. 1991; 19: 1-67.

Gunn SR. Support vector machines for classification and regression. Technical Report. 1998; University of Southampton.

MATLAB Basics. Matlab/Simulink tutorial. School of Electrical, Electronic and Computer Engineering. University of Newcastle Upon Tyne; 2003.

Montgomery DC. Design and analysis of experiments. New York: John Willey; 2001.

Myers RH, Montgomery DC. Response surface methodology: process and product optimization using designed experiments. New York: John Wiley; 1995.

Osuna E., Freund R, Girosi, F. An improved training algorithm for support vector machines: in neural networks for signal processing. Proceedings of the IEEE Workshop. New York. 1997; 276-285.

Osuolale KA, Yahya WB, Adeleke BL. Performing a simple pendulum experiment as a demonstrative example for computer experiments. Ann Comput Sci Ser. 2014a; 12: 33-38.

Osuolale KA, Yahya WB, Adeleke BL. Construction of space-filling designs for three input variables computer experiments. Int J Comput Control Quantum Inf Eng. 2014b; 8: 1624-1628.

Osuolale KA, Yahya WB, Adeleke BL. Modelling and analysis of simple pendulum computer experiment. Proceedings of the International Conference on Computer Science Research and Innovations (CoSRI 2015); 2015 Aug 21-22; Nigeria.University of Ibadan; 2015.

Osuolale KA, Yahya WB, Adeleke BL. Construction of orthogonal array-based Latin hypercube designs for deterministic computer experiments. Ann Comput Sci Ser. 2015b; 13(1): 24-29.

Osuolale KA. Modelling and analysis of simple pendulum computer experiment using a support vector regression model with B-splines kernel function. Proceedings of the International Conference on Applied Physics and Mathematics (ICAPM 2018). Flora Grand Hotel, Dubai, UAE; 2018

Strogatz S. The real scientific hero of 1953. New York: New York Times; 2003.

Scholkopf B, Burges C, Vapnik V. Incorporating invariances in support vector learning machines: In: von der Malsburg C, Seelen WV, Vorbruggen J, Sendhoff, B (Eds.), International Conference on Artificial Neural Networks (ICANN 1996). Berlin: Springer; 1996. p.47-52.