

Research Article

Optimizing the Prediction Accuracy of Friction Capacity of Driven Piles in Cohesive Soil Using a Novel Self-Tuning Least Squares Support Vector Machine

Doddy Prayogo ¹ and Yudas Tadeus Teddy Susanto^{1,2}

¹Department of Civil Engineering, Petra Christian University, Jalan Siwalankerto 121-131, Surabaya 60236, Indonesia

²PT. Sarana Data Persada, Margorejo Indah XIX/35-Blok D-507, Surabaya, Indonesia

Correspondence should be addressed to Doddy Prayogo; prayogo@petra.ac.id

Received 31 August 2017; Accepted 17 December 2017; Published 20 March 2018

Academic Editor: Andrea Benedetto

Copyright © 2018 Doddy Prayogo and Yudas Tadeus Teddy Susanto. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This research presents a novel hybrid prediction technique, namely, self-tuning least squares support vector machine (ST-LSSVM), to accurately model the friction capacity of driven piles in cohesive soil. The hybrid approach uses LS-SVM as a supervised-learning-based predictor to build an accurate input-output relationship of the dataset and SOS method to optimize the σ and γ parameters of the LS-SVM. Evaluation and investigation of the ST-LSSVM were conducted on 45 training data and 20 testing data of driven pile load tests that were compiled from previous studies. The prediction accuracy of the ST-LSSVM was then compared to other machine learning methods, namely, LS-SVM and BPNN, and was benchmarked with the previous results by neural network (NN) from Goh using coefficient of correlation (R), mean absolute error (MAE), and root mean square error (RMSE). The comparison showed that the ST-LSSVM performed better than LS-SVM, BPNN, and NN in terms of R , RMSE, and MAE. This comprehensive evaluation confirmed the capability of hybrid approach SOS and LS-SVM to modeling the accurate friction capacity of driven piles in clay. It makes for a reliable and robust assistance tool in helping all geotechnical engineers estimate friction pile capacity.

1. Introduction

Deep foundations built through the past years were made either of concrete, steel, or timber piles, which are either driven and precast or bored and cast-in-situ. Driven piles are frequently used in developing countries with a vast array of suburban and rural areas as foundations to support heavily loaded structures, for example, high-rise buildings and bridges. Recently, a variation of driven piles, jack-in piles, has also successfully used as foundations for high-rise buildings in urban areas due to their lower vibration and noise compared to the conventional-driven piles [1].

Despite all the development in the driven piles method, the design of driven piles still heavily relies on semi-empirical methods to estimate shaft resistance (f_s) and end resistance (f_b) [1–3]. The f_b will be negligible, and the large percentage of

pile load will be taken by f_s if there is no bearing layer, that is, driven piles in cohesive soil. Consequently, f_s that can be provided by the soil is very important in pile design. Up until now, there is no comprehensive assessment of f_s prediction methods [3]. Efforts were limited to comparing f_s prediction from various semiempirical methods with results from pile load tests [3–5]. To overcome this limitation, other efforts were dedicated to predicting f_s through machine learning techniques [6–8].

In civil engineering, machine learning techniques have developed into an important research area. Several studies reveal the advantages of using machine learning techniques in establishing a better predictive model over traditional methods [9–11]. Recently, least squares support vector machine (LS-SVM) has become one of the widely used machine learning techniques in handling variety of complex problems [12–15]. Although acceptable prediction results

have been reported, an improper parameter tuning may lessen the learning process of LS-SVM resulting in a lower accuracy. Building a more accurate predictive model can be achieved by optimizing the LS-SVM parameters. This includes a regularization parameter (γ) to deal with the trade-off between minimizing model complexity and training error. Also, it includes a kernel parameter (σ) of the radial basis function (RBF) which describes the nonlinear mapping between the input space and high-dimensional feature space.

Identifying the optimal parameters is an optimization issue, therefore, many recent studies combine a machine learning technique with a metaheuristic-based optimizer instead of using a sole machine learning [16–21]. Therefore, this research presents a new hybrid prediction method called self-tuning least squares support vector machine (ST-LSSVM) to accurately model the friction capacity of driven piles. The hybrid approach ST-LSSVM combines the techniques of SOS and LS-SVM. While the SOS is used to optimize the σ and γ parameters of the LS-SVM, the LS-SVM builds an accurate input-output relationship of the dataset by performing as a supervised-learning-based predictor. The total 45 training data and 20 testing data from Goh [6] have been employed to validate the performance of the proposed method. The ST-LSSVM method is further compared with LS-SVM and BPNN and is benchmarked with the previous results from Goh [6] using the coefficient of correlation (R), mean absolute error (MAE), and root mean square error (RMSE).

2. The Proposed Self-Optimized Machine Learning Framework

The objective of this proposed hybrid method is to improve the learning abilities of LS-SVM by searching for optimized set of LS-SVM parameters automatically. The collaborative integration between LS-SVM-based regression and SOS facilitates the LS-SVM to accurately determine the complicated relationship behavior between input variables and the output variable of the given historical data. The LS-SVM and SOS are briefly described below.

2.1. Machine Learning Technique: Least Squares Support Vector Machine (LS-SVM). LS-SVM is first introduced by Suykens and Vandewalle [12] as a modification of the conventional support vector machine (SVM). LS-SVM is used with a least squares loss function which allows for function estimation while reducing computational costs. Where highly nonlinear spaces occur, RBF kernel is chosen as the kernel function in LS-SVM which brings more promising results than other kernels [12, 22]. The following model of interest underlies the functional relationship between one or more independent variables along with a response variable [12, 23]:

$$y(x) = w^T \phi(x) + b, \quad (1)$$

where $x \in R^n$, $y \in R$, and $\phi(x) : R^n \rightarrow R^{nh}$ is the mapping to the high-dimensional feature space. In LS-SVM for regression

analysis, given a training dataset $\{x_k, y_k\}_{k=1}^N$, the optimization problem is formulated as follows:

$$\text{minimize } J_p(w, e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^N e_k^2,$$

$$\text{subjected to } y_k = w^T \phi(x_k) + b + e_k, \quad k = 1, \dots, N, \quad (2)$$

where $e_k \in R$ are the error variables and $\gamma > 0$ denotes the regularization constant.

In the previous optimization problem, a regularization term and a sum of squared fitting errors make for the objective function. For the cost function, this will be similar to the standard procedure with training feedforward neural networks and this is closely related to a ridge regression. However, the primal problem becomes somewhat impossible to solve when w becomes infinite dimensional. In this case, the dual problem should be derived after constructing the Lagrangian [12].

The Lagrangian is given by

$$L(w, b, e; \alpha) = J_p(w, e) - \sum_{k=1}^N \alpha_k \{w^T \phi(x_k) + b + e_k - y_k\}, \quad (3)$$

where α_k are the Lagrange multipliers. The conditions for optimality are given by

$$\frac{\partial L}{\partial w} = 0 \longrightarrow w = \sum_{k=1}^N \alpha_k \phi(x_k),$$

$$\frac{\partial L}{\partial b} = 0 \longrightarrow \sum_{k=1}^N \alpha_k = 0,$$

$$\frac{\partial L}{\partial e_k} = 0 \longrightarrow \alpha_k = \gamma e_k, \quad k = 1, \dots, N,$$

$$\frac{\partial L}{\partial \alpha_k} = 0 \longrightarrow w^T \phi(x_k) + b + e_k - y_k = 0, \quad k = 1, \dots, N. \quad (4)$$

After elimination of e and w , the following linear system is obtained:

$$\begin{bmatrix} 0 & \mathbf{1}_v^T \\ \mathbf{1}_v & \Omega + \left(\frac{I}{\gamma}\right) \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}, \quad (5)$$

where $y = [y_1, \dots, y_N]$, $\mathbf{1}_v = [1; \dots; 1]$, and $\alpha = [\alpha_1; \dots; \alpha_N]$. And the kernel function is applied as follows:

$$\Omega = \phi(x_k)^T \phi(x_l) = K(x_k, x_l). \quad (6)$$

The resulting LS-SVM model for function estimation is expressed as

$$y(x) = \sum_{k=1}^N \alpha_k K(x_k, x_l) + b, \quad (7)$$

where α_k and b are the solution to the linear system (5).



FIGURE 1: Symbiotic relationships (mutualism, commensalism, and parasitism) among organisms in an ecosystem.

The kernel function that is often utilized is RBF kernel. Description of RBF kernel is given as follows:

$$K(x_k, x_l) = \exp\left(-\frac{\|x_k - x_l\|^2}{2\sigma^2}\right), \quad (8)$$

where σ is the kernel function parameter.

With the γ parameter, the imposed penalty (to data points that move away from the regression function) can be controlled. For the σ parameter, this will have a direct impact on the smoothness of the regression function. To ensure the best performance of the predictive model, it should be noted that proper setting of these tuning hyperparameters is required.

2.2. Metaheuristic Optimization Algorithm: Symbiotic Organisms Search (SOS). Developed by Cheng and Prayogo, SOS is a recently developed metaheuristic algorithm that took inspiration from dependency-based interaction normally found among natural organisms and symbiosis [24]. Just like many other metaheuristic solutions, SOS guides the searching process using special operators that use candidate solutions; it looks for organisms containing candidate solutions to find the global solution in the search space; it requires a maximum number of evaluations and other common control parameters; and it preserves the better solutions by using a selection mechanism.

Nevertheless, there are some key differences because SOS does not need algorithm-specific parameters; for example, particle swarm optimization (PSO) relies upon the social factor, inertia weight, and cognitive factor. To tune the parameters, SOS requires no extra work, and this is a huge advantage. With improper tuning of the parameters, there is a possibility that the obtained solutions are found in local optima regions. Since the first development in 2014, SOS has been successfully utilized in solving many optimization problems in various research areas [25–31].

At the beginning, SOS will create a random ecosystem matrix (population) with each problem having a viable candidate solution. For the user, the number of organisms can be entered within the ecosystem, and this is called the ecosystem size. In each row of the matrix, this represents organisms which are the same as individuals in many other solutions. With each virtual organism, a candidate solution

is represented alongside the corresponding objective. Once the ecosystem has been generated, the search then begins.

With three clear phases, the idea comes from the most well-known symbioses. In the long term, organisms use them to improve their survival advantage and fitness (Figure 1). Throughout this searching process, there are three ways in which the organisms benefit from interacting with one another. The SOS algorithm adopts greedy selection scheme. Therefore, the updated organisms can replace the current organisms only if their fitness is better. Once one organism has finished all three phases, the best organism can then be updated. All things considered, the phases will form a continual cycle until the stopping criterion has been reached.

2.2.1. Mutualism Phase. With the mutualism phase, this is a relationship where the two sides benefit, and the perfect example would be flowers and bees. The mathematical formulation for the mutualism phase is shown as follows:

$$\text{new}O_i = \text{current}O_i + (\text{best}O - \text{mutual}O_{ij} * \text{BF}_1), \quad (9)$$

$$\text{new}O_j = \text{current}O_j + (\text{best}O - \text{mutual}O_{ij} * \text{BF}_2), \quad (10)$$

where $\text{current}O_i$ and $\text{current}O_j$ represent the two current organisms engaged in mutualism; $\text{best}O$ represents the current best organism; $\text{mutual}O_{ij}$ models the mutualism interaction between the two current organisms; $\text{new}O_i$ and $\text{new}O_j$ represent the updated organisms after the interaction; BF_1 and BF_2 are the two random values of either 0 or 1 representing the level of benefit of each organism. Meanwhile, $\text{mutual}O_j$ can be calculated using the following formulation:

$$\text{mutual}O_{ij} = \frac{(\text{current}O_i + \text{current}O_j)}{2}. \quad (11)$$

2.2.2. Commensalism Phase. In this phase, the organism manages to develop a relationship in which only they benefit. For example, this is common between sharks and remora fish. The mathematical formulation for the commensalism phase is shown as follows:

$$\text{new}O_i = \text{current}O_i + \text{rand}(-1, 1) * (\text{best}O - \text{current}O_j), \quad (12)$$

where $\text{rand}(-1, 1)$ is the uniform random parameter between -1 and 1 .

2.2.3. Parasitism Phase. Finally, this is a relationship where one side is benefitted and the other is harmed in some way. For instance, the plasmodium parasite transfers from one human host to the next using the *Anopheles* mosquito. In this phase, the beneficiary will get fitter while the harmed is likely to perish. The mathematical formulation for the parasitism phase is shown as follows:

$$\text{new}O_j = \text{parasite}O_i, \quad (13)$$

where $\text{parasite}O_i$ is the artificial parasite affiliated with $\text{current}O_i$ that threatens the existence of $\text{current}O_j$. Meanwhile, $\text{parasite}O_i$ can be calculated as follows:

$$\text{new}O_j = F * \text{current}O_i + (1 - F) * (\text{rand}(0, 1) * (\text{ub} - \text{lb})), \quad (14)$$

where F and $(1 - F)$ are the binary random matrix and its inverse, respectively, and $\text{rand}(0, 1)$ is the uniform random parameter between 0 and 1 .

2.3. Cross-Validation Technique and Performance Measurement. Training and testing processes are essential in establishing the prediction model. In the training process, a dataset is implemented to buildup a prediction model through the machine learning method. In the testing process, the established prediction model is used to validate new dataset. Using the entire dataset to train might cause an ‘‘overfitting’’ phenomenon, that is, the prediction model fits the dataset extremely well but is useless for a new and unseen dataset. Hence, the training dataset is often divided into two subsets to avoid the overfitting problem; larger portion of the training dataset as ‘‘training subset’’ and smaller portion of the training data as ‘‘validation subset.’’ The validation subset is used to validate the model built. This approach ensures the established prediction model to perform well in predicting the testing dataset.

To eliminate the randomness in partitioning the training dataset, k -fold cross-validation technique is employed [32, 33]. During this process, k -fold cross-validation creates k subsets from historical data, and they will always be nonoverlapping. The first $(k - 1)$ subset will be used in training the inference model which, in this case, is ST-LSSVM, before the last k th subset is used to validate the result. Since it relies upon cross-validation, this means that the process repeats itself for k times until all subsets have been used once as the validating subsets. At all times, k will remain as an unfixed parameter, and this means k can be any suitable number. In the current research, the value of k was set to 10 since this allowed computational time to minimize. This meant that all data were divided into 10 randomly ensuring equally sized groups. While one subset is used as testing data, the other nine can be used as training data. The term ‘‘tenfold’’ means that 10% of the data will be used as validating subsets while the remaining 90% is used as training subsets.

The performance measures used to evaluate the predictive methods are further described in Table 1. The

TABLE 1: Performance measurement for training dataset and testing dataset.

Performance measurement	Formula
Coefficient of correlation (R)	$R = \frac{n\sum y \cdot y' - (\sum y)(\sum y')}{\sqrt{n(\sum y^2) - (\sum y)^2} \sqrt{n(\sum y'^2) - (\sum y')^2}}$
Root mean square error (RMSE)	$\text{RMSE} = \sqrt{\left(\frac{1}{n}\right) \sum_{i=1}^n (y - y')^2}$
Mean absolute error (MAE)	$\text{MAE} = \left(\frac{1}{n}\right) \sum_{i=1}^n y - y' $

y is the actual value; y' is the predicted value; and n is the number of data samples.

performance measures are implemented on the predicted output results of the training and testing data. The lowest RMSE and MAE values, together with the highest R value, indicate the best model outcome.

2.4. Integration of LS-SVM and SOS with Cross-Validation. The procedure of ST-LSSVM explains the interaction of the proposed method in using training data and testing data to deliver the best prediction results. As mentioned earlier, the training dataset was divided as training subset and validating subset. In the training process, the building of predictive model sets was constructed by allowing SOS to determine the optimal LS-SVM parameters. Figure 2 presents the framework of the ST-LSSVM method.

To remove the issue of overfitting, k -fold cross-validation was chosen for parameter selection. From here, statistical performance measures can be used to calculate all results from the subset and folds. The best parameters represent the parameter set that can produce the minimum average RMSE on validating datasets through the tenfold training simulation. Meanwhile, the testing dataset focused on evaluating the performance of the trained LS-SVM model after finishing the optimization on unseen data. Using SOS, the whole optimization process was automated since this allowed for simultaneous optimization of LS-SVM. While SOS concentrated on optimizing the two LS-SVM parameters (γ and σ) to reduce prediction errors, LS-SVM addressed curve fitting and learning.

3. Experimental Results

3.1. Historical Dataset. Consisting of 45 training data and 20 testing data of load test records, this research uses historical data compiled by Goh [6]. With data-driven models mentioned here, they were based on the results received from various load data records for driven piles in clay including Vijayvergiya [34], Flaate and Selnes [35], and Semple and Rigden [36]; they were scaled down for laboratory conditions which means that the results should be used for similar conditions. In actual field data, it may not always fall within the range used in this study. Thus, dimensional analysis will need to be used alongside scaling effects to effectively apply the results here in actual field practice.

When expressing properties of load test records, numerous components were used including effective vertical

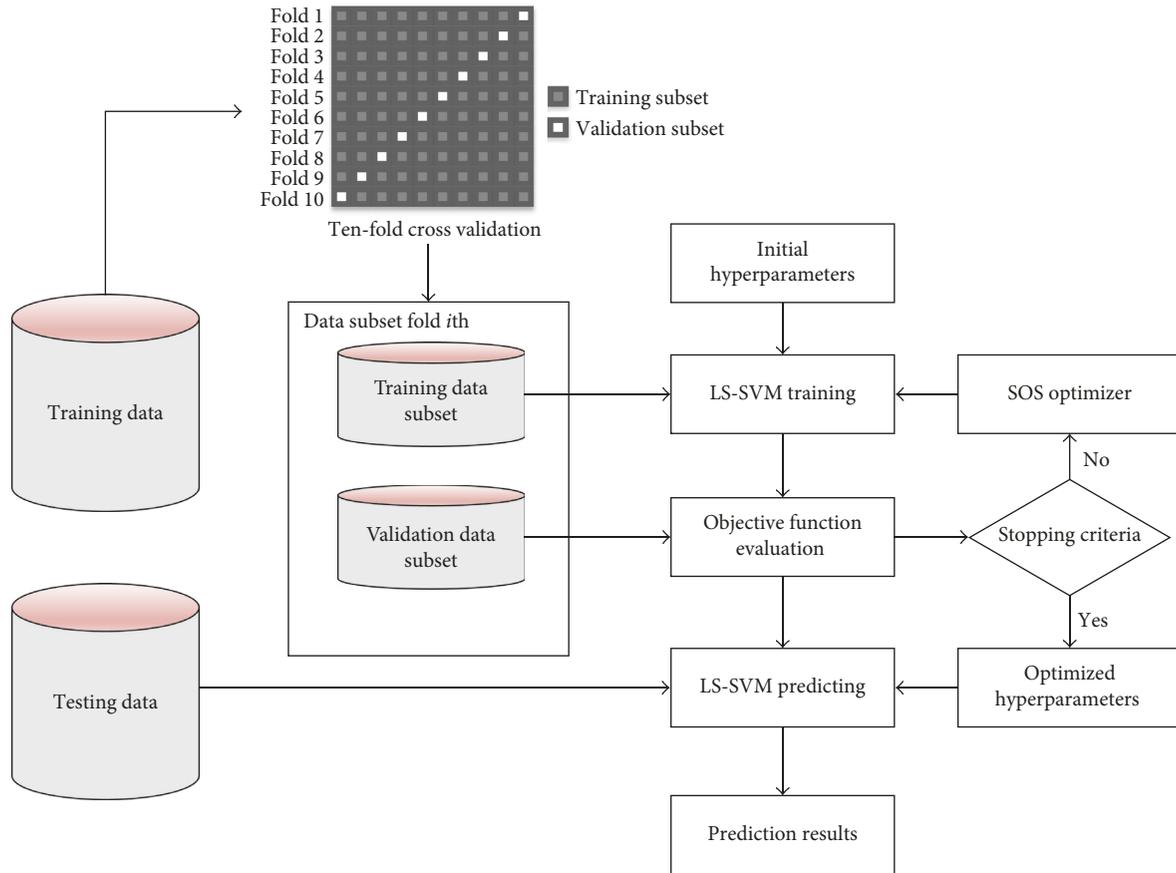


FIGURE 2: Framework of the ST-LSSVM method.

TABLE 2: Load test records properties and statistical descriptions.

Influencing factors	Unit	Training data				Testing data			
		Min	Max	Average	Std	Min	Max	Average	Std
Pile length (L)	m	4.60	96.00	21.56	17.22	8.00	66.40	21.55	14.70
Pile diameter (D)	cm	13.50	76.70	32.98	17.85	11.40	61.00	27.72	13.23
Undrained shear strength (σ'_v)	kPa	19.00	718.00	135.53	147.12	21.00	244.00	99.63	62.80
Effective vertical stress (S_u)	kPa	10.00	335.00	67.99	65.97	9.00	185.00	49.06	42.43
Friction capacity (f_s)	kPa	8.00	192.10	45.59	41.32	9.00	88.80	30.18	19.11

stress (kPa), pile length (m), undrained shear strength (kPa), and pile diameter (cm). It is worth noticing that the output of the load test is friction capacity (kPa). Statistical descriptions of training and testing datasets of load test records are reported in Table 2.

In the training dataset, the pile length varied from 4.6 meters to 96 meters, while the diameter started with a minimum of 13.5 cm and reached up to 76.7 cm. For effective vertical stress and undrained shear strength, these could be found from 19 to 718 kPa and 10 to 335 kPa, respectively. In terms of friction capacity, this started at 8 and reached 192.1 kPa. From the testing dataset, pile length and pile diameter were from 8 to 66.4 m and 11.4 to 61 cm, respectively. While the undrained shear strength started at 9 and reached 185 kPa, effective vertical stress was found between 21 and

244 kPa. Finally, measured friction capacity was between 9 and 88.8 kPa.

3.2. Training and Testing Processes. To build the pile capacity prediction model using previous data, the training process is essential. The k -fold cross-validation method can be used to ensure that the pile capacity model is as accurate as possible. The training process of ST-LSSVM with the given training dataset is simulated over 10 times based on cross-validation, with each 10 subset used exactly once as a validating subset. After the training process finished, the model is ready for predicting a new unseen testing dataset. The complete training result is provided in Appendix.

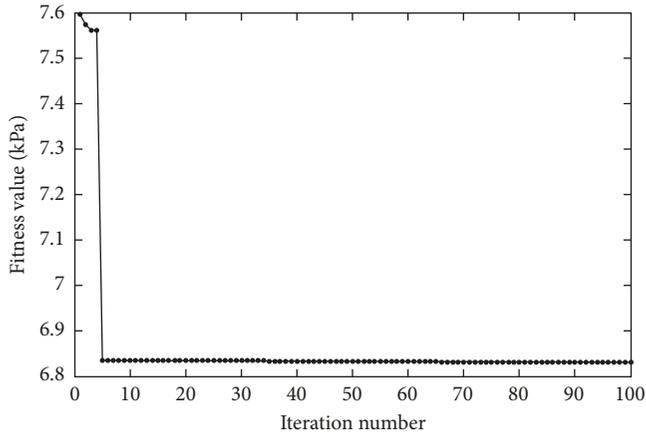


FIGURE 3: Convergence graphic of the training process.

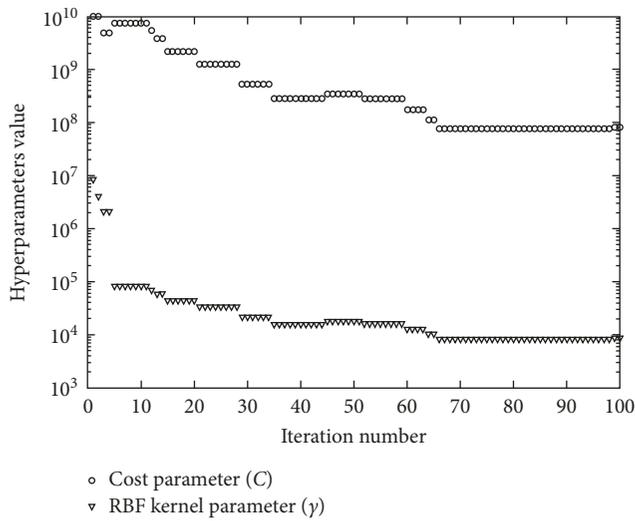


FIGURE 4: The iterative process of hyperparameters selection.

Through trial and error, suitable parameter settings for ST-LSSVM were determined: (1) maximum number of iterations is set to be 100, (2) population size is set to be 30, and (3) search range for the γ and σ^2 parameters is varied from 10^{-10} to 10^{10} as suggested in [37]. Using the given training dataset, the training procedure begins with random initial population of hyperparameters. For every iteration, ST-LSSVM simulates 10-fold cross-validation of training and validating subsets and stores the average RMSE value of the validating subset of each fold as the fitness value. The fitness value starts from a high RMSE value of 7.596 and iteratively decreases and converges to a RMSE value of 6.831 as shown in Figure 3. Figure 4 shows the historical records of the hyperparameters selection process. The final set of parameters that produce the lowest RMSE value on validation subsets are 81369676 for γ value and 8622 for σ^2 value. Finally, the complete testing result is provided in Appendix and evaluated using three above-mentioned performance measurements.

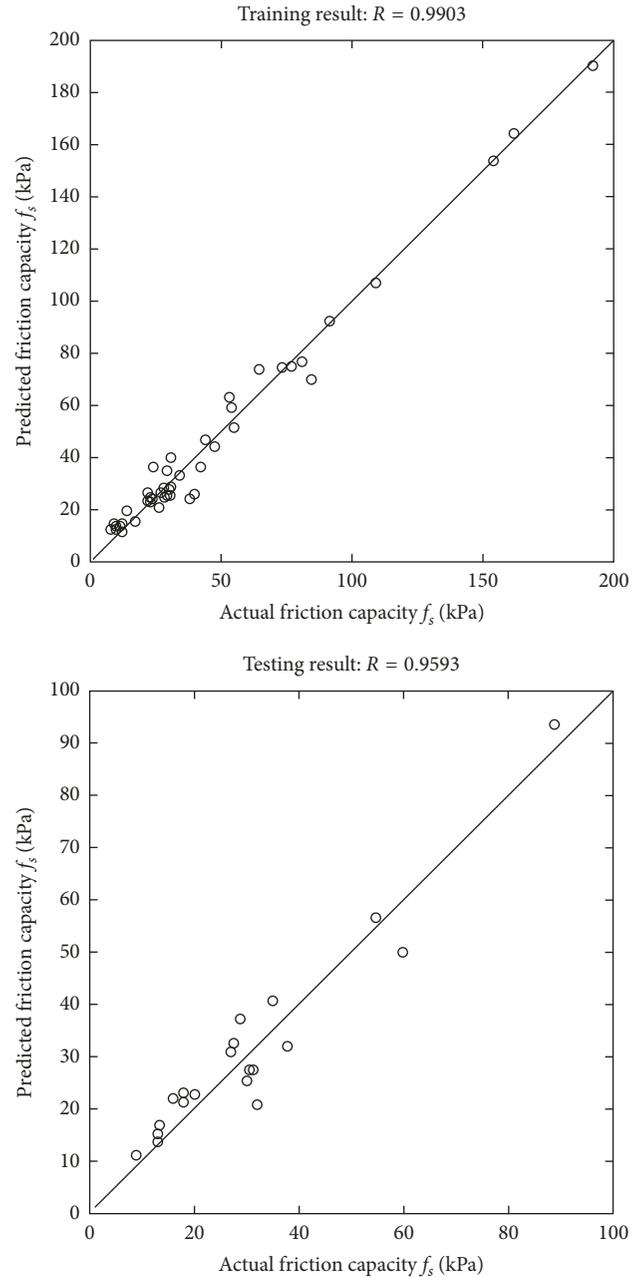


FIGURE 5: Obtained results of the ST-LSSVM predictive model for both training and testing datasets.

4. Results and Discussions

For comparison purposes, this research applied other machine learning-based predictive methods which are LS-SVM and backpropagation neural network (BPNN). The setting of BPNN follows the default parameter setting of MATLAB neural network toolbox, and Levenberg–Marquardt is chosen to train the BPNN [38]. The hyperparameters of LS-SVM follow the default setting suggested in the publication of Suykens and Vandewalle [12]. Additionally, the previous result from the literature published by Goh [6] is collected

TABLE 3: Comparative results of the predictive methods.

Methods	R value		RMSE value		MAE value		Hyperparameters
	Training	Testing	Training	Testing	Training	Testing	
Neural network [6]	0.985	0.956	6.824	5.333	5.111	4.715	N/A
Levenberg–Marquardt BPNN (present study)	0.9848	0.9480	7.6906	7.2189	6.255	6.5531	Final gradient = 205.2903, final $\mu = 100$
LS-SVM (present study)	0.9723	0.8802	13.9526	10.5474	9.351	7.7231	$\gamma = 1, \sigma^2 = 1$
ST-LSSVM (present study)	0.9903	0.9593	5.6678	5.3455	4.2381	4.6812	$\gamma = 81369675.72, \sigma^2 = 8622.39$

TABLE 4: Obtained prediction results by ST-LSSVM for training data.

L (m)	D (cm)	σ'_v (kPa)	S_u (kPa)	f_s (kPa)	Predicted f_s (kPa)
14.1	15	96	26	27	24.26
13	15	102	15	26	21.11
11.7	20	54	23	14	19.44
17.5	14.3	87	23	26	20.78
15.9	15	49	17	12	14.56
8.1	13.5	37	13	11	13.92
7.7	16.5	32	15	9	14.56
10	13.5	33	10	12	11.64
12	15.5	39	12	10	12.71
10.2	22	19	15	8	12.44
24.2	15	146	19	29	25.28
17.1	15	109	57	24	36.41
12.7	23.2	38	19	17	15.66
10	17	82	36	28	28.42
14.3	26	89	22	22	23.48
22.5	47	60	45	23	24.50
5.5	30.5	44	30	38	23.94
19.2	61	142	31	30.7	40.21
15.2	35.6	448	104	109.2	106.91
12.2	35.6	718	162	162	164.05
43.9	30.5	162	38	30	27.80
96	61	354	80	44	46.75
73.8	61	273	67	47.6	44.32
22.6	76.7	651	170	192.1	190.33
30.5	32.5	153	45	29.3	35.04
45.7	32.5	148	52	21.8	26.45
13.7	32.5	112	45	42.3	36.28
5.5	16.9	51.6	129.5	76.7	74.83
29	33	105	39	39.8	26.10
12.2	16.8	33	16	9.9	13.67
14	35.1	59	30	23.4	23.00
39.6	27.4	297	165	80.9	76.79
30.5	61	91	52	30.7	28.26
25.9	32.5	99	61	34.2	33.14
13.1	27.4	80	110	53.9	59.14
20.4	61	105	208	91.5	92.34
9.1	45	54	144	73.4	74.65
16.8	61	87	100	55	51.42
13.7	32.5	112	137	64.4	73.89

TABLE 4: Continued.

L (m)	D (cm)	σ'_v (kPa)	S_u (kPa)	f_s (kPa)	Predicted f_s (kPa)
18.3	76.2	115	335	154.1	153.65
4.6	16.9	43	120.5	84.6	70.20
33.6	32.5	121.4	35.4	30.4	25.65
33.6	32.5	108	48.8	27.1	26.39
20.3	32.5	158.2	112.8	53	63.12
30.5	51	102.8	24.4	23.5	24.02

for benchmarking the result obtained from the proposed method.

Figure 5 shows the obtained training and testing results of ST-LSSVM. The actual and predicted output of both training and testing phases showed a good fit to a straight line. The R -values for training and testing phases reported in this experiment also reflect the high accuracy and the superior performance of the trained ST-LSSVM model.

As mentioned earlier, to further express the accurate evaluation of the performing methods, RMSE and MAE have also been utilized, besides R . Table 3 compiles the prediction results of each method for further analysis. The results showed that the ST-LSSVM model effectively facilitated constructing an optimized predictive model over the default LS-SVM method. By implementing the self-optimized framework, the performance measures R , RMSE, and MAE of LS-SVM testing results were improved by 0.079, 5.2019, and 3.0419 kPa, respectively. Additionally, the results of ST-LSSVM are also superior to those of BPNN in terms of R , RMSE, and MAE. When comparing with neural network (NN) which was previously published by Goh [6], ST-LSSVM performs relatively better. It can be seen that ST-LSSVM achieves slightly better testing results over NN in two categories (R and MAE) while producing significantly better training results over NN in all categories. This comprehensive evaluation confirmed the capability of SOS and LS-SVM for modeling the accurate friction capacity based on load test records.

5. Conclusions

In this study, a new method for predicting friction pile capacity has been established based on load test records. With this research, it now extends the current body of knowledge that exists on investigating how capable LS-SVM

TABLE 5: Obtained predicted results by ST-LSSVM for testing data.

L (m)	D (cm)	σ'_v (kPa)	S_u (kPa)	f_s (kPa)	Predicted f_s (kPa)
8	13.5	27	9	9	11.11
9.4	29.3	52	29	18	23.08
14.6	16	67	29	16	21.96
11.6	17.5	57	27	18	21.28
9.6	19.2	42	15	13	15.30
21.6	45.7	147	31	28.8	37.22
36.9	30.6	149.6	28.2	30.5	27.39
66.4	32.5	223	60	31.2	27.43
11.6	11.4	44	21	13.4	16.86
22.9	32.5	91	52	27	30.92
13.8	19	21	21	13	13.78
25.3	27.4	244	185	88.8	93.59
14.9	52.8	66	53	27.6	32.57
18.3	32.5	51	33	32	20.83
48.2	61	152	64	37.8	31.93
32	21.4	141	115	59.8	49.98
13.4	27	81	22	20	22.76
24.2	15	147	19	30	25.42
15.5	17.5	80	72	35	40.60
12.8	32.5	110	96	54.7	56.55

is in predicting this information. Genuine load test records were collected, and the proposed model managed to achieve accurate prediction results. In this study, the main purpose was to investigate a hybrid computational intelligence system and its efficacy in optimizing the LS-SVM parameters to improve the accuracy of friction capacity forecasts of driven piles on the main load test records.

After analyzing further, the results suggested that LS-SVM, combined with SOS, could facilitate the construction of an optimized predictive model to be used for friction capacity. With modeling friction pile capacity being so complex and highly nonlinear, the obtained R , RMSE, and MAE values for both training and testing are impressive and desirable for most. With this new method, it makes for a reliable and robust assistance tool in helping all geotechnical engineers estimate friction pile capacity.

Appendix

Prediction Results by ST-LSSVM for Training and Testing Data

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] C. Chow and Y. Tan, "Jack-in pile design—Malaysian experience and design approach to EC7," *IEM Course on Eurocode*, vol. 7, 2009.

- [2] M. F. Randolph, "Science and empiricism in pile foundation design," *Géotechnique*, vol. 53, no. 10, pp. 847–875, 2003.
- [3] A. S. Azzouz, M. M. Baligh, and A. J. Whittle, "Shaft resistance of piles in clay," *Journal of Geotechnical Engineering*, vol. 116, no. 2, pp. 205–221, 1990.
- [4] M. Y. Abu-Farsakh and H. H. Titi, "Assessment of direct cone penetration test methods for predicting the ultimate capacity of friction driven piles," *Journal of Geotechnical and Geoenvironmental Engineering*, vol. 130, no. 9, pp. 935–944, 2004.
- [5] C. Roonnavaz, "Driven pile construction control procedures and design," *Proceedings of the Institution of Civil Engineers-Geotechnical Engineering*, vol. 163, no. 5, pp. 241–255, 2010.
- [6] A. Goh, "Empirical design in geotechnics using neural networks," *Geotechnique*, vol. 45, no. 4, pp. 709–714, 1995.
- [7] P. Samui, "Prediction of friction capacity of driven piles in clay using the support vector machine," *Canadian Geotechnical Journal*, vol. 45, no. 2, pp. 288–295, 2008.
- [8] S. Suman, S. K. Das, and R. Mohanty, "Prediction of friction capacity of driven piles in clay using artificial intelligence techniques," *International Journal of Geotechnical Engineering*, vol. 10, no. 5, pp. 469–475, 2016.
- [9] W. T. Chan, Y. K. Chow, and L. F. Liu, "Neural network: an alternative to pile driving formulas," *Computers and Geotechnics*, vol. 17, no. 2, pp. 135–156, 1995.
- [10] J.-S. Chou, C.-K. Chiu, M. Farfoura, and I. Al-Taharwa, "Optimizing the prediction accuracy of concrete compressive strength based on a comparison of data-mining techniques," *Journal of Computing in Civil Engineering*, vol. 25, no. 3, pp. 242–253, 2011.
- [11] N.-D. Hoang, A.-D. Pham, Q.-L. Nguyen, and Q.-N. Pham, "Estimating compressive strength of high performance concrete with Gaussian process regression model," *Advances in Civil Engineering*, vol. 2016, Article ID 2861380, 8 pages, 2016.
- [12] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.
- [13] P. Samui and D. P. Kothari, "Utilization of a least square support vector machine (LSSVM) for slope stability analysis," *Scientia Iranica*, vol. 18, no. 1, pp. 53–58, 2011.
- [14] P. Samui, "Least square support vector machine and relevance vector machine for evaluating seismic liquefaction potential using SPT," *Natural Hazards*, vol. 59, no. 2, pp. 811–822, 2011.
- [15] B. G. Aiyer, D. Kim, N. Karingattikkal, P. Samui, and P. R. Rao, "Prediction of compressive strength of self-compacting concrete using least square support vector machine and relevance vector machine," *KSCE Journal of Civil Engineering*, vol. 18, no. 6, pp. 1753–1758, 2014.
- [16] M.-Y. Cheng and D. Prayogo, "Modeling the permanent deformation behavior of asphalt mixtures using a novel hybrid computational intelligence," in *Proceedings of 33rd International Symposium on Automation and Robotics in Construction (ISARC'2016)*, pp. 1009–1015, International Association for Automation and Robotics in Construction, Auburn, AL, USA, July 2016.
- [17] M.-Y. Cheng, D. Prayogo, and Y.-W. Wu, "Novel genetic algorithm-based evolutionary support vector machine for optimizing high-performance concrete mixture," *Journal of Computing in Civil Engineering*, vol. 28, no. 4, p. 06014003, 2014.
- [18] J.-S. Chou and J. P. P. Thedja, "Metaheuristic optimization within machine learning-based classification system for early warnings related to geotechnical problems," *Automation in Construction*, vol. 68, pp. 65–80, 2016.

- [19] A.-D. Pham, N.-D. Hoang, and Q.-T. Nguyen, "Predicting compressive strength of high-performance concrete using metaheuristic-optimized least squares support vector regression," *Journal of Computing in Civil Engineering*, vol. 30, no. 3, p. 06015002, 2016.
- [20] M.-Y. Cheng, D. K. Wibowo, D. Prayogo, and A. F. V. Roy, "Predicting productivity loss caused by change orders using the evolutionary fuzzy support vector machine inference model," *Journal of Civil Engineering and Management*, vol. 21, no. 7, pp. 881–892, 2015.
- [21] M.-Y. Cheng, D. Prayogo, Y.-H. Ju, Y.-W. Wu, and S. Sutanto, "Optimizing mixture properties of biodiesel production using genetic algorithm-based evolutionary support vector machine," *International Journal of Green Energy*, vol. 13, no. 15, pp. 1599–1607, 2016.
- [22] J. A. Suykens, T. Van Gestel, and J. De Brabanter, *Least Squares Support Vector Machines*, World Scientific, Singapore, 2002.
- [23] J. A. K. Suykens, L. Lukas, and J. Vandewalle, "Sparse approximation using least squares support vector machines," in *Proceedings of 2000 IEEE International Symposium on Circuits and Systems. Emerging Technologies for the 21st Century*, pp. 757–760, Geneva, Switzerland, May 2000.
- [24] M.-Y. Cheng and D. Prayogo, "Symbiotic organisms search: a new metaheuristic optimization algorithm," *Computers & Structures*, vol. 139, pp. 98–112, 2014.
- [25] M.-Y. Cheng, D. Prayogo, and D.-H. Tran, "Optimizing multiple-resources leveling in multiple projects using discrete symbiotic organisms search," *Journal of Computing in Civil Engineering*, vol. 30, no. 3, p. 04015036, 2016.
- [26] D.-H. Tran, M.-Y. Cheng, and D. Prayogo, "A novel multiple objective symbiotic organisms search (MOSOS) for time-cost-labor utilization tradeoff problem," *Knowledge-Based Systems*, vol. 94, pp. 132–145, 2016.
- [27] M.-Y. Cheng, C.-K. Chiu, Y.-F. Chiu et al., "SOS optimization model for bridge life cycle risk evaluation and maintenance strategies," *Journal of the Chinese Institute of Civil and Hydraulic Engineering*, vol. 26, no. 4, pp. 293–308, 2014.
- [28] A. E.-S. Ezugwu, A. O. Adewumi, and M. E. Frincu, "Simulated annealing based symbiotic organisms search optimization algorithm for traveling salesman problem," *Expert Systems with Applications*, vol. 77, pp. 189–210, 2017.
- [29] A. Panda and S. Pani, "A Symbiotic Organisms Search algorithm with adaptive penalty function to solve multi-objective constrained optimization problems," *Applied Soft Computing*, vol. 46, pp. 344–360, 2016.
- [30] V. F. Yu, A. A. N. P. Redi, C.-L. Yang, E. Ruskartina, and B. Santosa, "Symbiotic organisms search and two solution representations for solving the capacitated vehicle routing problem," *Applied Soft Computing*, vol. 52, pp. 657–672, 2017.
- [31] H. Kamankesh, V. G. Agelidis, and A. Kavousi-Fard, "Optimal scheduling of renewable micro-grids considering plug-in hybrid electric vehicle charging demand," *Energy*, vol. 100, pp. 285–297, 2016.
- [32] P. Zhang, "Model selection via multifold cross validation," *Annals of Statistics*, vol. 21, no. 1, pp. 299–313, 1993.
- [33] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'1995)*, vol. 14, pp. 1137–1145, Montreal, QC, Canada, August 1995.
- [34] V. Vijayvergiya, "A new way to predict capacity of piles in clay," in *Proceedings of 4th Annual Offshore Technology Conference*, vol. 2, pp. 865–871, Houston, TX, USA, May 1972.
- [35] K. Flaate and P. Selnes, "Side friction of piles in clay," in *Proceedings of the 9th International Conference on Soil Mechanics and Foundation Engineering*, pp. 517–522, Tokyo, Japan, July 1977.
- [36] R. M. Semple and W. J. Rigden, "Shaft capacity of driven pipe piles in clay," in *Proceedings of Analysis and design of pile foundations (ASCE'1984)*, pp. 59–79, San Francisco, CA, USA, October 1984.
- [37] M.-Y. Cheng and N.-D. Hoang, "Risk score inference for bridge maintenance project using evolutionary fuzzy least squares support vector machine," *Journal of Computing in Civil Engineering*, vol. 28, no. 3, p. 04014003, 2014.
- [38] M. H. Beale, M. T. Hagan, and H. B. Demuth, *Neural Network Toolbox™ Reference*, MATLAB R2015b, The MathWorks, Natick, MA, USA, 1992.

