Soft Computing
https://doi.org/10.1007/s00500-018-3400-x

METHODOLOGIES AND APPLICATION

A self-tuning least squares support vector machine for estimating the pavement rutting behavior of asphalt mixtures

Min-Yuan Cheng[1] · Doddy Prayogo[2] · Yu-Wei Wu[1]

Abstract
The present study proposes a new self-tuning least squares support vector machine, called MSOS-SVM, for modeling the pavement rutting behavior of asphalt mixtures. MSOS-SVM combines the least squares support vector machine (LS-SVM), the symbiotic organisms search (SOS), and chaotic maps. In this system, the LS-SVM is used to establish the relationship model between the flow number obtained from laboratory tests and the parameters specified in the asphalt mix design. SOS is used to find the best LS-SVM tuning parameters. Meanwhile, chaotic system is used to enhance the exploration and exploitation process of SOS. A total of 118 historical cases were used to establish the intelligence-prediction model. The results validate the ability of MSOS-SVM to model the pavement rutting behavior of asphalt mixtures to a relatively high level of accuracy as measured using four error indicators. The present study demonstrates that the proposed computational intelligence system is a highly beneficial decision-making tool for road designers and engineers.

Keywords Asphalt mixtures · Computational intelligence · Prediction · Rutting behavior · Least squares support vector machine · Symbiotic organisms search

1 Introduction

The field of pavement engineering has focused increasing attention on the pavement rutting behavior of asphalt mixtures in recent decades. Vehicle load cycles are positively related to rutting in asphalt pavements, which degrades the service life of this type of pavement and creates hazardous conditions for roadway users (Sousa et al. 1991). Furthermore, this rutting reduces pavement thickness and reduces useful pavement life through fatigue cracking, decreased drainage capacity, water pooling, and uneven road surfaces. Repetitive vehicle traffic has been shown to be a main cause of pavement rutting and of the permanent deformation of asphalt roadways (Kaloush 2001). Therefore, understanding the permanent deformation behavior of asphalt mixtures under repeated loading and formulating proper asphalt mixtures prior to installation on roadways are critical to creating asphalt road surfaces that offer optimal durability and safety for roadway users (Alavi et al. 2011). The dynamic creep test is one of the best methods currently known for assessing the permanent deformation potential of asphalt mixtures (Kaloush et al. 2002). The "flow number", the number of loading cycles at which permanent deformation starts (Witczak et al. 2002), is a good indicator of the rutting resistance of a given asphalt mixture. However, the complexity and cost of the dynamic creep test make this test infeasible for widespread, frequent use. Therefore, a relationship model, which uses historical data and matches the flow numbers obtained from the dynamic creep test to the related parameters obtained from the asphalt mix design, was developed to provide a significantly less expensive, less time consuming, and sufficiently accurate alternative to the dynamic creep test.

Predicting the performance of asphalt mixtures is a critical part of roadway planning and construction. However, building a relationship model that accurately describes asphalt mixture performance has traditionally been a complicated task as the permanent deformation of asphalt mixtures is influenced by several factors that are complex and highly nonlinear. Conventional approaches, such as linear regres-

Communicated by V. Loia.

✉ Doddy Prayogo
prayogo@petra.ac.id

[1] Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan

[2] Department of Civil Engineering, Petra Christian University, Surabaya, Indonesia

Published online: 19 July 2018

sion and decision tress, are inadequate in terms of accuracy and computational cost for building a satisfactory model. Advanced artificial intelligence (AI) approaches have been reported to outperform the conventional approaches based on their excellent learning features in a wide variety of research fields (Kartelj et al. 2014; Cheng et al. 2014a; Cheng and Prayogo 2016; Sartakhti et al. 2017), including pavement design and engineering (Ahmed et al. 2017; Gu et al. 2018; Amin and Amador-Jiménez 2017). Nevertheless, only few studies have used AI to predict the flow number of asphalt mixtures (Alavi et al. 2011; Gandomi et al. 2010; Mirzahosseini et al. 2011).

The primary objective of the present research work is to build an accurate prediction model for permanent deformation that uses advanced AI methods to enhance the analysis of pavement rutting behavior. A new technique, the modified symbiotic organisms search-least squares support vector machine (MSOS-SVM), is proposed for this purpose. The MSOS-SVM combines a prediction technique, the least squares support vector machine (LS-SVM) (Suykens and Vandewalle 1999), with an optimization tool, symbiotic organisms search (SOS) (Cheng et al. 2014b), and a pseudo-randomness element of a chaotic system. To the best of our knowledge, this paper is the first to study the integration between three AI elements, chaotic system, SOS, and LS-SVM, for predicting pavement rutting behavior. LS-SVM is used to create a prediction model to find the relationship between rutting behavior and asphalt mixtures, SOS is used to optimize the hyperparameters of LS-SVM to increase the prediction accuracy, and chaotic system is used to enhance the exploration and exploitation process of SOS.

The proposed model is investigated alongside other prediction methods in terms of its efficacy as an accurate prediction model for pavement rutting in asphalt mixtures. Furthermore, a cross-validation technique is used to validate the training and test process and four different measures of accuracy are used to evaluate the performance of each prediction method.

2 Literature review

Numerous studies in the field of pavement engineering have in recent decades proposed approaches for modeling the rutting/permanent deformation potential of asphalt mixes. Conventional permanent deformation models have primarily used empirical models, which have limited material characterization abilities that lead to poor correlations with the actual performance of asphalt mixes in the field. There are three categories of rutting evaluation methods: (1) mechanistic-empirical modeling, (2) advanced constitutive modeling, and (3) simple tests performed during the design phase to evaluate the fundamental engineering properties and responses of mixture materials (Kim 2008). Several studies have aimed to identify the most effective models (Kaloush 2001; Kenis 1977; NCHRP 2004; Kannemeyer and Visser 1995). However, the accuracy of mechanistic-empirical rutting models is directly correlated with the quality and quantity of the empirical data employed in the calibration. Another significant area of research on this topic has focused on the viscoelastic, viscoplastic, and damage response components impacting the behavior of asphalt concrete directly. As a result of these efforts, three-dimensional, nonlinear, finite element codes were created (Gibson 2006). However, this finite, element-based approach has certain limitations, e.g., its high case-to-case variability results in poor generalizability. Additionally, a high level of prior knowledge about the nature of a relationship between variables is required by this approach.

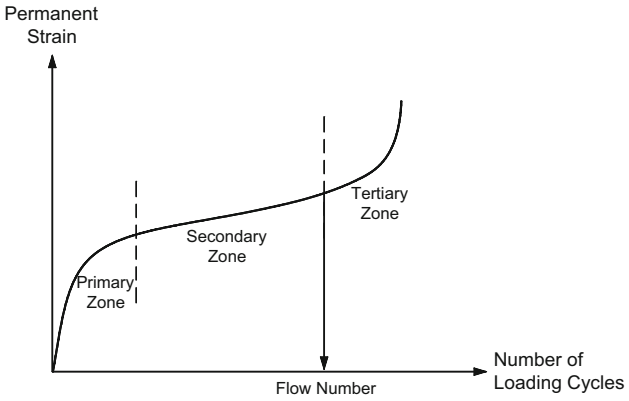

Fig. 1 Plot of accumulated strain versus number of loading cycles

The dynamic creep test is one of the best methods currently available to assess the rutting potential of asphalt mixtures (Kaloush et al. 2002). This test employs data from a few thousand repetitions of a repeated load test to record the cumulative permanent deformation. Permanent deformation is the main cause of rutting on asphalt mixtures. As shown in Fig. 1, according to Witczak et al. (2002), and illustrated in Fig. 1, the flow number is defined as the number of loading cycles at which permanent (tertiary) deformation begins. Thus, the flow number can be used to measure the resistance of the mixture to permanent deformations (Christopher et al. 2007).

Moreover, each test-experimental site report presents all of the repeated load permanent deformation data, allowing comparisons to be made between these parameters and rutting. In summary, of the many mixture-response parameters that correlate well with the measured rut depths, flow number was the highest-ranked test parameter, comparing well with the measured rut depths for all projects and test sections. Hence, a model evaluating the relationship between the flow number acquired from the dynamic creep test and the param-

eters obtained from mixture designs enables designers and road engineers to evaluate pavement rutting behavior efficiently, considering cost and time.

In one of the earliest studies conducted to measure the rutting potential of asphalt mixtures, Gandomi et al. (2010) used flow number as the target output predictor. The AI gene expression programming (GEP) method was used to build an accurate prediction model using a total of 118 samples from a real-world historical dataset. This model was subsequently benchmarked using multivariate least squares regression (MLSR) analysis, with results supporting the model as an effective tool for measuring the flow number based on the asphalt pavement mixture designs.

Using the same historical dataset, Alavi et al. (2011) used the hybrid genetic programming and simulated annealing (GP/SA) method to build a prediction model for asphalt mixture performance. The dataset was randomly divided, with 89 data points (75%) used as training set and 29 data points (25%) used as test set. Sensitivity analysis was used to evaluate the effects of the variables on the flow number.

To assess the rutting potential of dense asphalt-aggregate mixtures, Mirzahosseini et al. (2011) employed two artificial neural network models, multi-expression programming (MEP) and multilayer perceptron (MLP). Subsequently, MLSR analysis was used to benchmark these two models. Moreover, Yan et al. (2014) investigated the capacity of the support vector machine (SVM) for predicting the flow number based on the asphalt pavement mixture designs. The results were compared with results acquired by MLSR and GEP.

Previous studies in this area have made limited use of AI techniques. Furthermore, these studies have used only a simple, random division of training and test sets in the validation process. A more advanced validation method is necessary to eliminate the potential for bias in dividing data points between these two sets. Thus, more reliable and advanced models are needed to predict the flow number of asphalt mixtures.

3 Methodology

3.1 Prediction method: LS-SVM

LS-SVM is a modified version of SVM (Suykens and Vandewalle 1999), and LS-SVM is a statistical learning theory that adopts a least squares linear system as a loss function rather than as a quadratic program as in original SVM (Kulkrni et al. 2011). The LS-SVM is a new machine learning method offering numerous advanced features for fast computation and good generalization. As confirmed by empirical studies, LS-SVM is at least equally as accurate as conventional SVM; nonetheless, the computing efficiency of LS-SVM is higher (van Gestel et al. 2004). Furthermore, an LS-SVM in a machine learning model uses training samples and not the identification of supporting vectors as does conventional SVM. The following formula expresses the optimization problem and the constraints for LS-SVM:

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

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

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

The resulting LS-SVM model for function estimation may be expressed as:

$$y(x) = \sum_{k=1}^{N} \alpha_k K(x_k, x_l) + b, \tag{3}$$

where α_k and b are the solutions to the linear system (4).

The radial basis function (RBF) kernel is the kernel function that is most frequently used. The RBF may be expressed as:

$$K(x_k, x_l) = \exp\left(-\frac{\|x_k - x_l^2\|}{2\sigma^2}\right), \tag{4}$$

where σ is the kernel function parameter.

3.2 Optimization method: modified SOS

3.2.1 Basic symbiotic organisms search

SOS is a newly promising metaheuristic algorithm first proposed by Cheng et al. (2014b) and that has been used extensively to solve many types of engineering problems (Prayogo et al. 2017; Tran et al. 2016; Cheng et al. 2016; Yu et al. 2017). As shown in Fig. 2, SOS simulates three symbiotic interactions through the iterative moving of an ecosystem (population) of organisms (candidate solutions) toward better areas during the process of finding the optimal global solution. All organisms have a certain fitness value. The fitness value reflects the objective value, which corresponds to the candidate solution.

The following explains the three phases of mutualism symbiosis, commensalism symbiosis, and parasitism symbiosis, which simulate the three types of symbiotic interactions that occur in the real world.

```
Initialize an ecosystem of organisms x
while the stopping criterion is not satisfied
    for i = 1 to ecosystem size do
        Update the current best solution x_best
        Simulate mutualism symbiosis between x_i and x_ii
        Simulate commensalism symbiosis between x_i and x_ii
        Simulate parasitism symbiosis between x_i and x_ii
    end for
end while
```

Fig. 2 Pseudo-code of the SOS algorithm

In the mutualism symbiosis phase, the interactions of an organism with another organism are mutually beneficial. The following equation expresses this phase:

$$x_{i\,\text{new}} = x_i + \text{rand}\,(0,1) * \left[x_{\text{best}} - \left(\frac{x_i + x_{ii}}{2}\right) * (1 + \text{round}\,(\text{rand}\,(0,1)))\right], \tag{5}$$

$$x_{ii\,\text{new}} = x_{ii} + \text{rand}\,(0,1) * \left[x_{\text{best}} - \left(\frac{x_i + x_{ii}}{2}\right) * (1 + \text{round}\,(\text{rand}\,(0,1)))\right], \tag{6}$$

$$x_i = \begin{cases} x_i & f\,(x_i) \le f\,(x_{i\,\text{new}}) \\ x_{i\,\text{new}} & f\,(x_i) > f\,(x_{i\,\text{new}}) \end{cases}, \tag{7}$$

$$x_{ii} = \begin{cases} x_{ii} & f\,(x_{ii}) \le f\,(x_{ii\,\text{new}}) \\ x_{ii\,\text{new}} & f\,(x_{ii}) > f\,(x_{ii\,\text{new}}) \end{cases}, \tag{8}$$

where x_i is the ith organism vector of the ecosystem, x_{ii} is the iith organism vector of the ecosystem in which $ii \neq i$, x_{best} represents the best organism in the current generation, $x_{i\,\text{new}}$ and $x_{ii\,\text{new}}$ represent the respective candidate solutions for x_i and x_{ii} after their interaction, $f(x_i)$ is the fitness value of x_i, $f(x_{ii})$ is the fitness value of x_{ii}, $f(x_{i\,\text{new}})$ is the fitness value of $x_{i\,\text{new}}$, and $f(x_{ii\,\text{new}})$ is the fitness value of $x_{ii\,\text{new}}$.

In the commensalism symbiosis phase, the interactions of an organism with another organism benefit that organism and have no effect on the other organism. The following equation expresses this phase:

$$x_{i\,\text{new}} = x_i + \text{rand}(-1,1) * (x_{\text{best}} - x_{ii}), \tag{9}$$

$$x_i = \begin{cases} x_i & f(x_i) \le f(x_{i\,\text{new}}) \\ x_{i\,\text{new}} & f(x_i) > f(x_{i\,\text{new}}) \end{cases}, \tag{10}$$

where x_i is the ith organism vector of the ecosystem, x_{ii} is the iith organism vector of the ecosystem in which $ii \neq i$, x_{best} represents the best organism in the current generation, $x_{i\,\text{new}}$ represents the candidate solutions for x_i after the interaction, $f(x_i)$ is the fitness value of x_i, and $f(x_{i\,\text{new}})$ is the fitness value of $x_{i\,\text{new}}$.

In the parasitism symbiosis phase, the interactions of an organism with another organism benefit that organism and harm the other organism. The following equation expresses this phase:

$$x_{\text{parasite}} = \begin{cases} x_i & \text{if rand}\,(0,1) \le \text{rand}\,(0,1) \\ \text{LB} + \text{rand}\,(0,1) * (\text{UB} - \text{LB}) & \text{if rand}\,(0,1) > \text{rand}\,(0,1) \end{cases}, \tag{11}$$

$$x_{ii} = \begin{cases} x_{ii} & f\,(x_{ii}) \le f\,(x_{\text{parasite}}) \\ x_{\text{parasite}} & f\,(x_{ii}) > f\,(x_{\text{parasite}}) \end{cases}, \tag{12}$$

where x_i is the ith organism vector of the ecosystem, x_{ii} is the iith organism vector of the ecosystem in which $ii \neq i$, x_{parasite} is the artificial parasite organism created to compete with the host organism x_{ii}, $f(x_{ii})$ is the fitness value of x_{ii}, $f(x_{\text{parasite}})$ is the fitness value of x_{parasite}, LB is the lower bound of the problem, and UB is the upper bound of the problem.

3.2.2 Chaotic maps for manipulating the exploitation and exploration of SOS

Due to the stochastic nature of metaheuristic algorithms, it is not possible to set a clear boundary between exploration and exploitation during the search process. There is a possibility of metaheuristic algorithms to be trapped in local optima with a lack of adequate balance between exploration and exploitation. Accordingly, an increasing number of studies have aimed to enhance the performance of metaheuristic algorithms through the improvement of exploration and exploitation.

The past decade has seen increasing interest in the application of chaotic system to improve exploration and exploitation of metaheuristic algorithms. A chaos map is a set of functions that describe some sort of random and chaotic behavior, but not necessarily random. Thus, a chaos map proves that a deterministic system can also exhibit random behavior (Saremi et al. 2014). Zhenyu et al. (2006) applied a chaotic mutation factor to enhance the performance of differential evolution (DE). In the study of Gandomi et al. (2013), the global search of the firefly algorithm (FA) was increased by using chaotic maps instead of parameters of FA. Moreover, the chaotic maps were integrated by Saremi et al. (2014) into the selection, emigration, and mutation probabilities of biogeography-based optimization (BBO).

All mentioned studies confirmed the capacity to enhance the performance of metaheuristic algorithms. As a result, this study aims to manipulate the exploration and exploitation in the standard SOS phases by applying the chaotic mapping operator in place of random parameters. In this study,

Table 1 Chaotic maps

No.	Chaotic maps	Formulation
1	Iterative	$C_{t+1} = \sin\left(\frac{a\pi}{C_t}\right), a = 0.7$
2	Logistic	$C_{t+1} = aC_t(1 - C_t), a = 4$
3	Sinusoidal	$C_{t+1} = aC_t^2 \sin(\pi C_t), a = 2.3$

the chaotic maps used are investigated and the method of SOS performance enhancement through the chaotic maps is further explained. Table 1 and Fig. 3 show three selected chaotic maps (Saremi et al. 2014). All chaotic values (C_t) were mapped in the interval between 0 and 1. Following (Gandomi et al. 2013), 0.7 was used as the initial point for all. It is also worth pointing out that Fig. 3 shows quite clear chaotic behaviors, while Table 1 does not show random components.

This study used chaotic maps to manipulate the mutualism and commensalism operators of the SOS algorithm. As can be seen in Eqs. (5) and (6), the modification of the organism in the mutualism symbiosis phase is influenced by the uniform random parameter of rand(0, 1). A large value generated by rand(0, 1) enables the exploration of a new and promising region, but it takes a long time for the organisms to converge. On the other hand, a small value generated by rand(0, 1) facilitates exploitation but it sometimes leads organisms to prematurely converge on local optima. Chaotic maps were used to replace the uniform random parameter as follows:

$$x_{i\,\text{new}} = x_i + C_t * \left[x_{\text{best}} - \left(\frac{x_i + x_{ii}}{2}\right) * (1 + \text{round}\,(\text{rand}\,(0, 1)))\right], \tag{13}$$

$$x_{ii\,\text{new}} = x_{ii} + C_t * \left[x_{\text{best}} - \left(\frac{x_i + x_{ii}}{2}\right) * (1 + \text{round}\,(\text{rand}\,(0, 1)))\right]. \tag{14}$$

Similarly, according to Eq. (9), the modification of organisms in the commensalism symbiosis phase is affected by the uniform random parameter of rand(−1, 1). The chaotic maps were used to replace the uniform random parameter as follows:

$$x_{i\text{new}} = x_i + C_t * (x_{\text{best}} - x_{ii}). \tag{15}$$

In the modified SOS algorithm, different chaotic maps for mutualism symbiosis and commensalism symbiosis provide

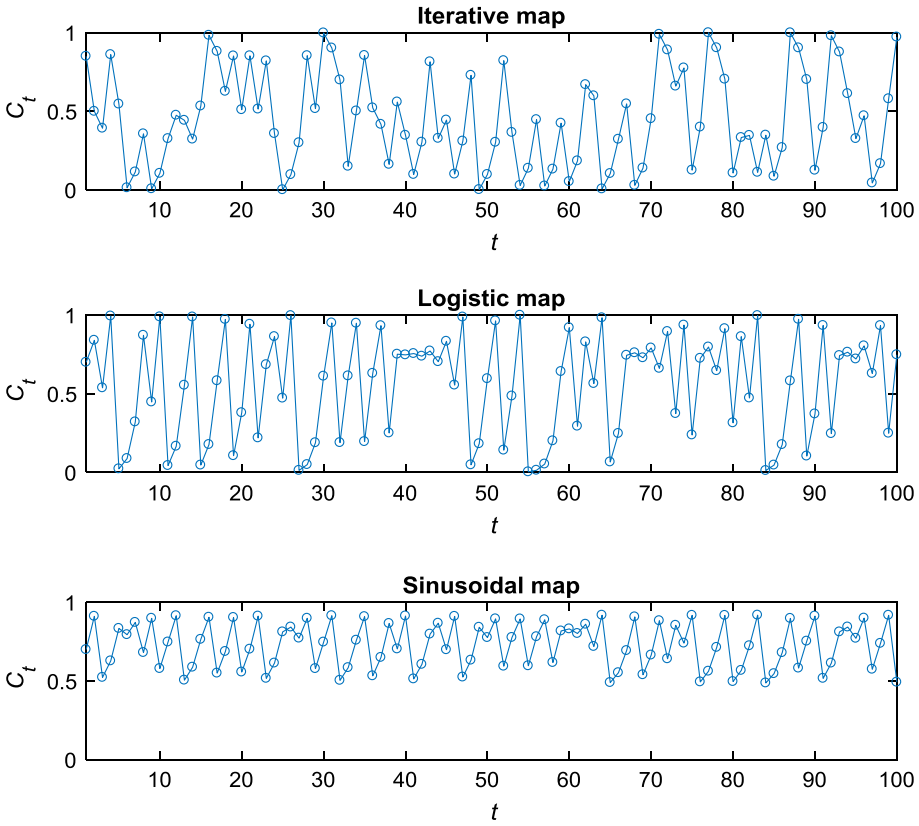

Fig. 3 Illustration of chaotic behavior for iterative, logistic, and sinusoidal maps

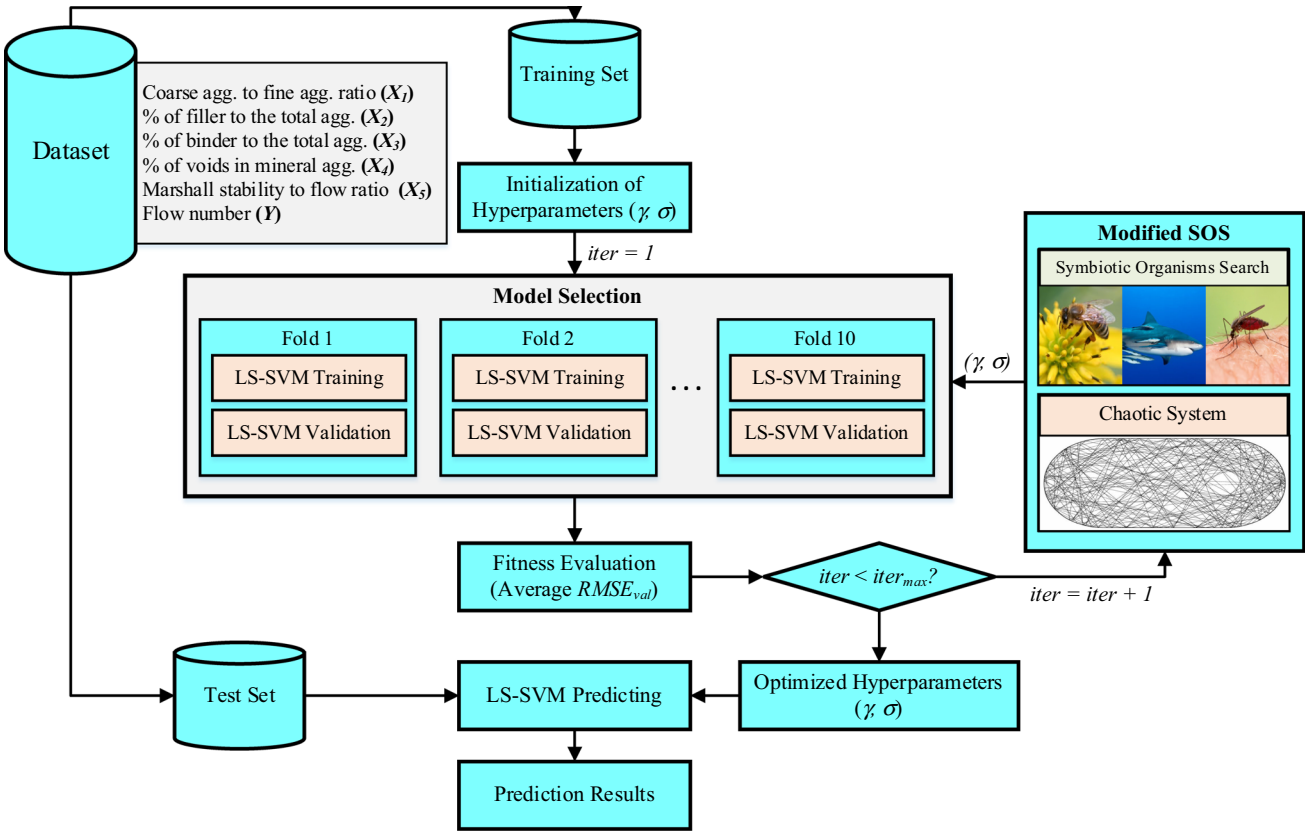

Fig. 4 MSOS-SVM architecture

different searching patterns for exploration and exploitation. A symbiotic interaction of SOS gives priority either to exploration or exploitation, considering that a chaotic map exhibits chaotic behavior.

3.3 MSOS-SVM system integration

This study combines the several different AI techniques of SOS, chaotic system, and LS-SVM in a hybrid AI system called MSOS-SVM. The LS-SVM plays an important role as a predictor that accurately maps the relationship of input and output variables of the given dataset. The SOS and chaos system are utilized to optimize the LS-SVM parameters γ and σ. Figure 4 shows the architecture of MSOS-SVM.

The six main steps of the SOS-SVM are conducted across the training and test phases and are explained below:

1. Dataset, training set, and test set: The dataset is divided into a training set (70%) and a test set (30%). Furthermore, to avoid the situation when one or some input variables dominate others, the datasets were scaled into a (0,1) range (Hsu et al. 2003). The training set is further divided into training subset and validation subset for model selection through cross-validation. The test set is used to measure the performance of the LS-SVM prediction model.
2. Initialization of hyperparameters:
In the first iteration, the parameters are initialized randomly within the boundary range using the following formula:

$$x = \text{rand}(0, 1) * (\text{UB} - \text{LB}) + \text{LB}, \tag{16}$$

where x represents candidate solution (hyperparameters), UB represents upper bound, and LB represents lower bound. For the present study, the lower and upper bounds were set to 10^{-10} and 10^{10}, respectively.
3. Model selection and fitness evaluation: This step is a critical and important step for building the accurate learning model. LS-SVM model is trained using the training set and the initial hyperparameters with a goal to find the accurate relationship between input and output variables. LS-SVM requires two parameters to operate, γ and σ, to conduct the learning process. The training process is conducted iteratively, and the tuning parameters are gradually optimized using the MSOS algorithm. To evaluate the accuracy of the learning system, a fitness function that is correlated with the accuracy of the prediction model is now developed. The combination of γ and σ parameters

Table 2 Statistical description of permanent deformation analysis obtained from the dynamic creep test

Attribute	Unit	Type	Minimum	Maximum	Average	SD
C/S	–	Input	0.58	4.5	1.8410	1.0506
FP	%	Input	1	10	5.5424	3.1721
BP	%	Input	4	7	5.5085	0.8138
VMA	%	Input	13.2	19.04	16.5513	1.4123
M/F	–	Input	0.61	4.81	2.9872	0.7425
Fn	–	Output	22	510	227	143.9741

Table 3 Training results by SOS and MSOS over the tenfolds

Optimizer	Optimal fitness value (average validation RMSE)	Hyperparameters	
		Final γ	Final σ^2
SOS	34.6147	1.8356E+09	4.3391E+03
$MSOS_1$	34.5491	8.1941E+08	3.4134E+03
$MSOS_2$	34.4811	2.2330E+08	2.1421E+03
$MSOS_3$	34.4799	2.3992E+08	2.2888E+03

that produces the best fitness value is considered as the most accurate prediction model. In the training process, the potential exists for the system to fit a poor training dataset on the new dataset, which is a problem known as "over-fitting" (Bishop 2006). To overcome this problem, a well-known technique, k-fold cross-validation, is incorporated in the fitness function. The dataset is now split into k folds, which assigns the $(k-1)/k$ portion of the dataset for training and assigns the remaining portion for validating the prediction model. A total of k-distinct sets of training and validation subset are performed. The fitness function uses root-mean-square error (*RMSE*) to represent model accuracy as shown in Eq. (17):

$$\text{fit\_val} = \frac{\sum_{k=1}^{S} \text{RMSE}_{\text{val}}}{S}, \tag{17}$$

where S is the total number of folds and fit_val is the fitness value calculated from the RMSE between the predicted and actual values for the validation set.

4. Modified SOS for parameter search: The hybrid AI system uses MSOS to explore the various combinations of γ and σ parameters to identify the best set of these hyperparameters. The search process commences with the generation of the initial population that represents the initial candidate solution for the searched hyperparameters. As mentioned before, the population is then encoded into continuous variables with the boundary limit set from 0 to 1. MSOS utilizes the mutualism, commensalism, and parasitism phases with the help of a chaotic system for each iteration to gradually improve the fitness value of each candidate solution in the population.
5. Optimal hyperparameters: The loop stops when the termination criterion is fulfilled, which implies that the prediction model has identified the input/output mapping relationship with optimal γ and σ parameters.
6. LS-SVM predicting: The optimal LS-SVM γ and σ parameters that were obtained from the training phase are used to establish the prediction model for predicting the test set.

4 Experimental results

4.1 Data collection and preparation

The study used an experimental dataset acquired from Alavi et al. (2011). A set of 118 experiment samples of dynamic creep tests was collected from the Asphalt Mixtures and Bitumen Research Center at the Iran University of Science and Technology. The purpose of the uniaxial dynamic creep test was to detect the flow number, which correlates with the rutting performance the asphalt mixtures. The details of aggregates, fillers, and bitumen characteristics are provided in Alavi et al. (2011). The experimental dataset of five input variables and one output variable employed in this study is presented in Table 2. The ratio of coarse aggregate to fine aggregate (C/S), percentage of filler to the total aggregate (FP), percentage of binder to the total aggregate (BP), percentage of voids in mineral aggregate (VMA), ratio of Marshall stability to Marshall flow (M/F), and flow number (Fn) are the main attributes of the dataset. "Appendix" provides the whole dataset of the dynamic creep test samples.

4.2 Model selection and training results

In this study, the performance of three MSOSs in selecting the optimal hyperparameters was benchmarked with the original SOS. As mentioned previously, three types of chaos maps utilized by MSOSs were investigated: iterative map ($MSOS_1$), logistic map ($MSOS_2$), and sinusoidal map ($MSOS_3$). To ensure a fair comparison, the MSOSs and SOS used the same

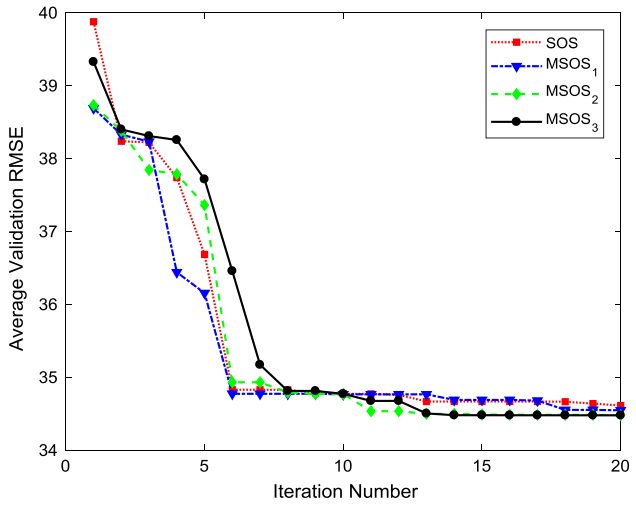

Fig. 5 Convergence history of the average validation error for each algorithm

settings: maximum number of iterations = 20 and ecosystem size = 20. The boundary range for the γ and σ parameters was set between 10^{-10} and 10^{10}.

The tenfold cross-validation splits the training set into 10 sets of different training and validation subsets. The model selection was performed by MSOSs and SOS using 10 sets of training and validation subsets, and the average validation errors were used as the fitness values. Table 3 shows the training performance of each model. Figure 5 shows the convergence history of the model selection using MSOSs and SOS. It can be seen from Fig. 5 and Table 3 that all variants of MSOS are able to find the lower validation error in comparison with the original SOS. $MSOS_2$ and $MSOS_3$ can yield the most optimum fitness value, slightly better than the other models. Furthermore, Fig. 6 shows the hyperparameters selection history of each model.

4.3 Prediction results

This research used four performance metrics to evaluate the prediction accuracy of the proposed method and other prediction techniques as follows:

1. Correlation coefficient (R)
   R is a statistical measure of how well a regression line approximates the real data points. The following equation is used to express R:

$$R = \frac{n\sum y.y' - \left(\sum y\right)\left(\sum y'\right)}{\sqrt{n\left(\sum y^2\right) - \left(\sum y\right)^2}\sqrt{n\left(\sum y'^2\right) - \left(\sum y'\right)^2}}. \tag{18}$$

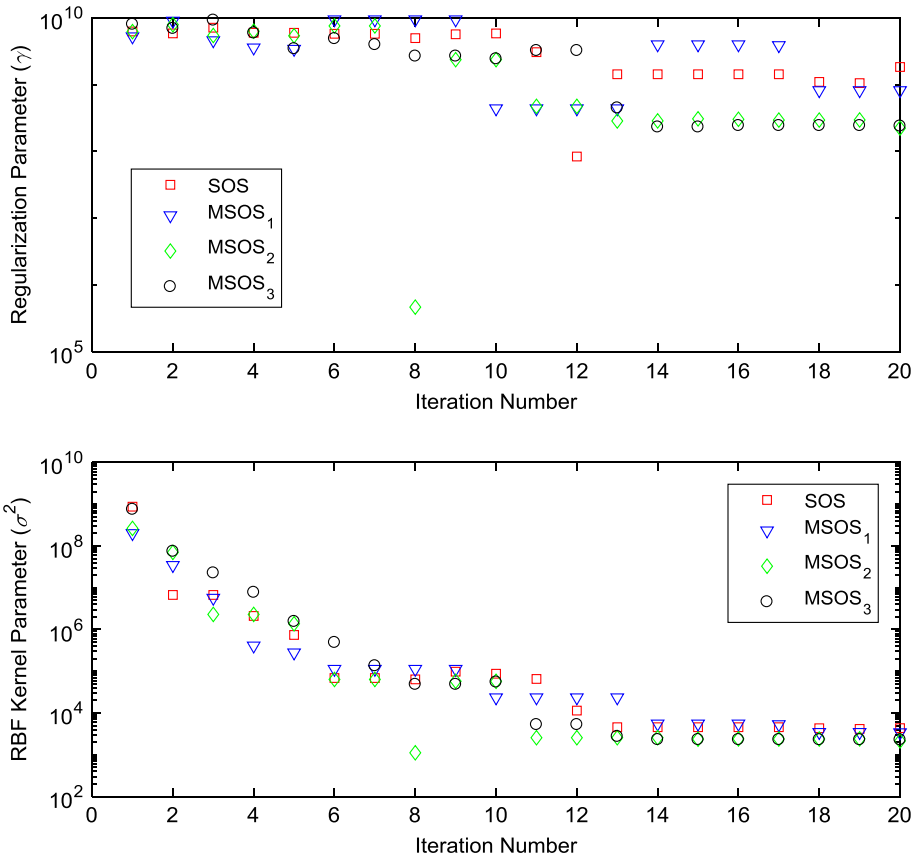

Fig. 6 Hyperparameters selection history for each algorithm

Table 4 Test performance of MSOS-SVMs and other methods

AI methods	Performance indicators			
	R	RMSE	MAE	MAPE
BPNN	0.9582	49.2929	32.3731	0.2500
LS-SVM	0.9459	72.5298	58.7483	0.7979
SOS-SVM	0.9712	38.7542	24.5113	0.2828
$MSOS_1$-SVM	0.9720	38.2396	23.9216	0.2469
$MSOS_2$-SVM	0.9724	37.9132	23.4834	0.2347
$MSOS_3$-SVM	0.9723	38.0390	23.5934	0.2424

Table 5 Details of test results of MSOS-SVMs and other methods

Test	Actual	Deviation between actual and predicted Fn					
Case	Fn	BPNN	LS-SVM	SOS-SVM	$MSOS_1$-SVM	$MSOS_2$-SVM	$MSOS_3$-SVM
16	55	11.08	42.73	1.23	1.24	1.26	1.07
17	50	0.47	54.85	6.21	4.10	3.96	4.25
18	300	26.31	12.94	39.61	40.60	40.62	40.39
21	300	38.37	17.89	57.03	60.34	59.65	58.64
28	370	52.85	85.07	76.75	78.86	78.39	77.87
32	310	7.82	23.50	23.03	25.23	24.91	24.30
38	190	35.80	21.51	5.00	4.20	3.44	3.69
40	300	14.75	25.03	4.47	6.57	6.00	5.33
46	300	57.56	14.22	21.34	21.20	21.24	21.35
47	510	153.50	212.25	151.76	152.99	153.67	153.75
52	38	8.74	44.66	10.34	10.27	10.26	10.16
54	500	130.18	103.23	28.69	30.72	30.27	29.63
56	60	7.00	56.68	5.40	5.33	5.62	5.48
57	420	79.63	88.40	5.48	3.58	2.74	3.38
58	24	37.51	91.96	54.52	44.99	41.94	43.55
59	60	26.60	31.60	9.83	7.26	7.22	7.79
64	40	6.62	51.93	6.23	3.41	1.71	2.10
66	380	27.91	120.49	28.12	28.94	28.67	28.55
69	52	12.11	36.24	13.25	14.23	14.05	13.87
71	380	9.49	46.22	10.29	8.38	8.47	9.07
73	230	33.76	64.27	40.19	39.37	40.18	40.71
82	60	20.43	51.28	2.37	1.25	1.33	1.49
83	50	0.50	63.62	3.69	3.43	2.85	3.21
87	50	14.40	44.83	3.76	4.12	4.05	3.81
90	300	23.97	43.61	41.66	44.01	44.52	43.86
92	320	8.47	36.86	3.27	4.69	4.19	3.67
94	150	90.06	14.51	5.42	7.70	7.75	7.39
95	60	1.44	60.90	12.25	11.53	11.82	12.07
98	370	4.96	68.03	41.85	42.71	40.88	40.64
101	310	16.37	15.38	24.03	25.07	24.96	24.57
102	60	8.92	46.49	13.84	13.40	13.40	13.56
103	38	8.64	45.55	11.53	7.69	6.92	7.52
105	22	39.00	165.14	78.70	65.25	61.47	65.34
113	480	111.37	113.16	5.09	3.90	4.64	4.95
118	40	6.47	41.16	11.68	10.68	8.89	8.75

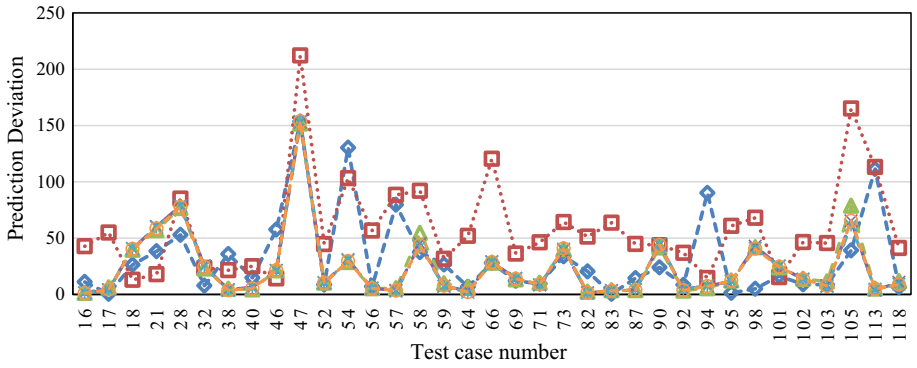

Fig. 7 Prediction deviation of LS-SVM, BPNN, SOS-SVM, and MSOS-SVMs for the test set

2. Root-mean-square error (RMSE)
   RMSE is the square root of the average squared distance between the values that are predicted by the model and the values that are observed. RMSE may be used to calculate the variation of errors in a prediction model and is very useful when large errors are undesirable. The following equation is used to express RMSE:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}, \tag{19}$$

   where y_j is the actual value, $\hat{y}_j$ is the predicted value, and n is the number of data samples.
3. Mean absolute error (MAE)
   MAE is the average absolute value of the residual (error). MAE is used to measure the closeness of forecasts or predictions to the actual outcomes. The following equation is used to express MAE:

$$\text{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j| \tag{20}$$

4. Mean absolute percentage error (MAPE)
   MAPE is used to measure prediction accuracy in terms of prediction percentage error. Small denominators cause problems in the MAPE value as these values generate large MAPE values that impact overall value. The following equation is used to express MAPE:

$$\text{MAPE} = \frac{1}{n}\sum_{j=1}^{n}\left|\frac{y_j - \hat{y}_j}{y_j}\right| \times 100\% \tag{21}$$

To validate the performance of the developed MSOS-SVM models, comparisons with other predictive models, including the original LS-SVM and back-propagation neural network (BPNN), in addition to SOS-SVM, were performed. The comparison between MSOS-SVM and other predictive algorithms may imply the advantages of using the optimization method to tune the optimal parameters. BPNN settings included: maximum hidden layers = 1; number of neurons in the hidden layer = 5 (equal to the number of input variables); and learning rate = 1. Finally, the LS-SVM parameters for σ and γ were set to 1 as suggested in Suykens and Vandewalle (1999).

Table 4 shows the complete set of experimental results. These results show that the three MSOS-SVM models performed better in the overall measurement category. In terms of *R*, *RMSE*, *MAE*, and *MAPE*, the MSOS-SVMs earned the best score followed by the SOS-SVM. The LS-SVM earned the worst scores for prediction ability. MSOS_2-SVM has produced a slightly better performance among the MSOS-SVM models.

The detailed results for test set of the MSOS-SVMs, SOS-SVM, LS-SVM, and BPNN, respectively, are displayed in Table 5. Figure 7 further illustrates the test deviations between the actual and predicted values for the three prediction techniques. The three proposed MSOS-SVM models produced the best data fit among the prediction techniques that were evaluated, further supporting that the MSOS-SVM is the most reliable algorithm for establishing the prediction model.

4.4 Comparison with previous works

Many studies have been conducted to predict the rutting behavior of asphalt mixtures using AI methods. As mentioned earlier in Sect. 2, some previous works that have performed the rutting behavior modeling of the asphalt mixtures are: Gandomi et al. (2010) with the gene expression programming (GEP); Alavi et al. (2011) with the hybrid genetic programming and simulated annealing (GP/SA) method; and Mirzahosseini et al. (2011) with the multi-expression programming (MEP). Table 6 compares the test performance between the MSOS-SVM and other predictive techniques. It is worth noting that the data partitioning for training and test sets might be different between the present study and the past work because the data partitioning is often not shown in the past literature. As shown in Table 6, all MSOS-SVM models perform better in different performance metrics.

Table 6 Test performance of MSOS-SVMs and other methods

AI methods	Ref	Performance metrics		
		R	RMSE	MAE
MEP	Mirzahosseini et al. (2011)	0.956	46.23	32.509
GP/SA	Alavi et al. (2011)	0.948	46.06	33.842
GEP	Gandomi et al. (2010)	0.891	67.63	48.218
$MSOS_1$-SVM	Present study	0.9720	38.2396	23.9216
$MSOS_2$-SVM	Present study	**0.9724**	**37.9132**	**23.4834**
$MSOS_3$-SVM	Present study	0.9723	38.0390	23.5934

Bold indicates best performance

5 Conclusion

The present study developed a new self-tuning prediction method called the modified symbiotic organisms search-least squares support vector machine (MSOS-SVM) to predict permanent deformation in asphalt mixtures. SOS is a very promising metaheuristic algorithm that offers important advantages over traditional metaheuristic algorithms, including less control parameters. The three unique phases of SOS were shown to cover the search space of the hyperparameters effectively, which reduced the risk of entrapment in local optima. The chaotic maps are utilized to further enhance the SOS performance in searching the optimal hyperparameters. Furthermore, three prediction methods, SOS-SVM, LS-SVM, and BPNN, were used as a benchmark for the MSOS-SVM. The experimental dataset was acquired from a prior dynamic creep test of 118 samples.

Furthermore, four performance metrics (MAPE, MAE, RMSE, and *R*) were used to additionally compare the proposed MSOS-SVM for performance outcomes to quantitatively compare in detail various predictive techniques. According to the results, the most accurate performance measure is the proposed MSOS-SVM, with the SOS-SVM, BPNN, and LS-SVM achieving the second-, third-, and fourth-best overall accuracies, respectively. It is obvious that the original LS-SVM was outperformed by the hybrid MSOS-SVM because of the success of the MSOS in finding parameters of better fit than the default parameter settings in the LS-SVM. The superiority of MSOS-SVM over BPNN indicates that the proposed methods perform better than other, currently prevalent prediction methods. Meanwhile, the superiority of the MSOS-SVM over SOS-SVM reveals the success of chaos maps in improving the exploration and exploitation of the basic SOS.

This study conclusively established that the new predictive model, MSOS-SVM, enables road planners and engineers to resolve a crucial problem of pavement rutting in asphalt mixtures. The innovative model predicts precisely the proper flow number of asphalt mixtures, and accordingly designers are enabled to choose mixtures that meet specifications. The findings proved that the MSOS-SVM is the optimal model for developing asphalt mixtures with particular permanent deformation characteristics.

Compliance with ethical standards

Conflict of interest No potential conflict of interest was reported by the authors.

Appendix

See Table 7.

Table 7 Dynamic creep test conducted on the asphalt mix samples

No.	Input					Output
	C/S	FP (%)	BP (%)	VMA (%)	M/F	Fn
1	1.45	7	5	15.12	3.39	340
2	2.27	2	5.5	16.38	3.37	50
3	1.16	6	5.5	17.38	3.68	230
4	1.45	7	4	16.3	3.59	260
5	2.27	2	6.5	17.51	2.84	60
6	1.45	7	4	16.16	3.27	350
7	0.88	10	5.5	16.69	4.17	440
8	4.5	1	4.5	13.7	2.35	37
9	1.16	6	7	18.32	2.68	240
10	2.43	4	5.5	15.43	2.33	180
11	1.16	6	6.5	17.74	3.28	260
12	0.58	10	6.5	17.88	4.33	400
13	2.43	4	5.5	15.13	2.49	200
14	0.58	10	7	18.82	2.56	350
15	1.16	6	5	17.96	2.97	160
16	2.27	2	6	17.05	2.86	55
17	2.27	2	5.5	16.36	2.45	50
18	1.45	7	4.5	15.45	3.41	300
19	1.16	6	6	17.08	3.58	270
20	0.58	10	5.5	18.24	2.41	380
21	0.58	10	5.5	18.15	2.32	300
22	2.43	4	4	14.05	3.51	170

Table 7 continued

No.	Input					Output
	C/S	FP (%)	BP (%)	VMA (%)	M/F	Fn
23	2.77	2	4.5	15.29	3.09	80
24	0.88	10	6.5	17.92	3.45	480
25	1.45	7	6	16.99	2.65	280
26	2.43	4	5	14.72	2.74	210
27	1.16	6	7	18.54	2.99	220
28	1.54	6	6	15.94	3.18	370
29	0.88	10	4.5	18.22	3.63	340
30	0.88	10	6	17.05	3.98	500
31	2.77	2	4.5	14.95	3.86	48
32	1.45	7	4.5	15.51	3.07	310
33	0.58	10	5	18.98	1.81	320
34	2.43	4	4.5	14.31	3.39	160
35	2.77	2	5.5	15.8	2.66	80
36	1.16	6	5	18.13	3.3	180
37	2.27	2	7	17.97	2.45	45
38	2.43	4	4.5	14.54	3.3	190
39	2.27	2	6	16.83	2.84	55
40	1.54	6	5	15.65	3.72	300
41	1.45	7	6	17.42	2.86	240
42	1.54	6	4.5	16.26	3.24	290
43	1.54	6	4.5	15.8	3.41	260
44	2.43	4	5	14.92	2.82	150
45	1.16	6	5.5	17.63	3.27	270
46	0.58	10	5	18.87	1.77	300
47	0.88	10	4.5	18.29	2.93	510
48	1.16	6	5	18.25	3.23	210
49	2.77	2	6.5	17.23	1.76	50
50	2.27	2	7	18.49	2.57	50
51	2.43	4	5.5	15.65	2.41	140
52	4.5	1	4.5	13.76	2.03	38
53	2.77	2	5	15.25	3.29	75
54	0.88	10	5.5	16.71	4.32	500
55	2.27	2	6.5	17.21	2.84	60
56	2.77	2	4.5	15.24	2.98	60
57	0.88	10	5	17.1	4.81	420
58	4.5	1	5.5	14.33	1.46	24
59	2.77	2	6.5	17.06	1.75	60
60	1.45	7	5.5	15.92	3.44	350
61	2.27	2	5.5	16.64	3.15	45
62	0.88	10	6	17.48	4.56	420
63	0.88	10	4.5	17.99	4.13	400

Table 7 continued

No.	Input					Output
	C/S	FP (%)	BP (%)	VMA (%)	M/F	Fn
64	4.5	1	4	13.56	2.29	40
65	0.58	10	5.5	18.35	2.5	340
66	1.45	7	5	14.99	4.16	380
67	0.58	10	7	18.46	2.7	360
68	1.54	6	5	15.43	3.99	340
69	2.77	2	6	16.3	2.17	52
70	2.77	2	6	16.25	2.29	65
71	0.58	10	6	17.93	3.78	380
72	0.58	10	6.5	18.46	3.47	380
73	1.54	6	4.5	16.25	4.02	230
74	2.43	4	6	17.55	1.15	160
75	1.54	6	5.5	16.14	2.77	380
76	0.88	10	5	16.86	3.99	450
77	0.88	10	5	17.03	4.07	460
78	0.88	10	6.5	17.98	3.83	440
79	4.5	1	4	13.2	1.88	38
80	2.27	2	5	16.38	3.61	40
81	1.45	7	5.5	15.1	2.95	265
82	2.27	2	6	16.55	2.9	60
83	2.27	2	5	16.22	2.88	50
84	2.43	4	4	14.19	3.27	160
85	0.58	10	5	18.94	1.85	320
86	1.54	6	6.5	17.17	2.84	280
87	2.27	2	6.5	17.59	2.62	50
88	1.16	6	6	17.13	4.06	220
89	2.43	4	6	15.09	2.79	160
90	1.16	6	6	17.49	3.16	300
91	1.54	6	6	16.82	2.75	350
92	1.45	7	5.5	15.95	2.67	320
93	0.88	10	6	16.94	3.54	450
94	2.43	4	6	15.46	2.03	150
95	2.77	2	5	15.24	3.34	60
96	2.27	2	5	16.32	2.44	45
97	2.77	2	5	15.29	2.83	65
98	0.58	10	7	19.04	2.29	370
99	2.77	2	5.5	15.72	2.82	44
100	0.58	10	6	18.19	3.25	340
101	1.45	7	5	15.54	3.05	310
102	2.77	2	5.5	15.82	2.58	60
103	4.5	1	4	13.75	2.05	38
104	1.16	6	6.5	17.79	3.01	260
105	4.5	1	6	16.22	0.61	22

Table 7 continued

No.	Input					Output
	C/S	FP (%)	BP (%)	VMA (%)	M/F	Fn
106	0.58	10	6	17.99	3.52	320
107	4.5	1	5	14.9	1.79	36
108	2.77	2	6.5	17.34	1.91	44
109	1.16	6	6.5	17.81	3.45	240
110	1.54	6	6.5	17.23	2.66	320
111	0.58	10	6.5	17.86	3.89	400
112	1.54	6	5.5	15.92	3.08	370
113	0.88	10	5.5	16.25	4.41	480
114	1.16	6	7	18.89	2.77	250
115	2.43	4	5	14.52	2.63	200
116	1.45	7	6	15.5	2.37	250
117	1.54	6	5.5	16.22	3.52	380
118	4.5	1	4.5	14.2	2.26	40

References

Ahmed TM, Green PL, Khalid HA (2017) Predicting fatigue performance of hot mix asphalt using artificial neural networks. Road Mater Pavement Des 18:141–154

Alavi AH, Ameri M, Gandomi AH, Mirzahosseini MR (2011) Formulation of flow number of asphalt mixes using a hybrid computational method. Constr Build Mater 25(3):1338–1355

Amin SR, Amador-Jiménez LE (2017) Backpropagation neural network to estimate pavement performance: dealing with measurement errors. Road Mater Pavement Des 18(5):1218–1238

Bishop CM (2006) Pattern recognition and machine learning (information science and statistics). Springer, New York

Cheng M-Y, Prayogo D (2016) Modeling the permanent deformation behavior of asphalt mixtures using a novel hybrid computational intelligence. Paper presented at the ISARC 2016—33rd international symposium on automation and robotics in construction, Auburn, USA

Cheng M-Y, Prayogo D, Wu Y-W (2014a) Novel genetic algorithm-based evolutionary support vector machine for optimizing high-performance concrete mixture. J Comput Civ Eng 28(4):06014003

Cheng M-Y, Firdausi PM, Prayogo D (2014b) High-performance concrete compressive strength prediction using Genetic Weighted Pyramid Operation Tree (GWPOT). Eng Appl Artif Intel 29:104–113

Cheng MY, Prayogo D, Tran DH (2016) Optimizing multiple-resources leveling in multiple projects using discrete symbiotic organisms search. J Comput Civ Eng 30(3):04015036

Christopher WR, Robinette CJ, Bausano J, Breakah T (2007) Testing of Wisconsin asphalt mixtures for the forthcoming AASHTO 2002 mechanistic-empirical pavement design procedure. Wisconsin Highway Research Program

Gandomi A, Alavi A, Mirzahosseini M, Nejad F (2010) Nonlinear genetic-based models for prediction of flow number of asphalt mixtures. J Mater Civ Eng 23(3):248–263

Gandomi AH, Yang XS, Talatahari S, Alavi AH (2013) Firefly algorithm with chaos. Commun Nonlinear Sci Numer Simul 18(1):89–98

Gibson NH (2006) A viscoelastoplastic continuum damage model for the compressive behavior of asphalt concrete. Ph.D. dissertation, University of Maryland, College Park

Gu F, Luo X, Zhang Y, Chen Y, Luo R, Lytton RL (2018) Prediction of geogrid-reinforced flexible pavement performance using artificial neural network approach. Road Mater Pavement Des 19(5):1147–1163

Hsu CW, Chang CC, Lin CJ (2003) A practical guide to support vector classification. Technical report, Department of Computer Science and Information Engineering, National Taiwan University. Available at http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

Kaloush KE (2001) Simple performance test for permanent deformation of asphalt mixtures. Ph.D. Thesis, Arizona State University

Kaloush KE, Witczak MW et al (2002) Tertiary flow characteristics of asphalt mixtures. J Assoc Asphalt Paving Technol 71:278–306

Kannemeyer L, Visser AT (1995) Calibration of HDM-III performance models for use in pavement management of South African national roads. Transp Res Rec 1508:31

Kartelj A, Mitić N, Filipović V, Tošić D (2014) Electromagnetism-like algorithm for support vector machine parameter tuning. Soft Comput 18(10):1985–1998

Kenis WJ (1977) Predictive design procedures–a design method for flexible pavements using the VESYS structural subsystem. In: Proceedings of 4th international conference on structural design of asphalt pavements, vol 1. University of Michigan, Ann Arbor, Michigan, pp 101–130

Kim YR (2008) Modeling of asphalt concrete, 1st edn. McGraw-Hill, New York

Kulkrni KS, Kim D-K, Sekar SK, Samui P (2011) Model of least square support vector machine (LSSVM) for prediction of fracture parameters of concrete. Int J Concrete Struct Mater 5(1):29–33

Mirzahosseini MR, Aghaeifar A, Alavi AH, Gandomi AH, Seyednour R (2011) Permanent deformation analysis of asphalt mixtures using soft computing techniques. Expert Syst Appl 38(5):6081–6100

NCHRP (2004) Development of the 2002 guide for the design of new and rehabilitated pavement structures: phase II. NCHRP 1-37A, American Association of State Highway and Transportation Officials, USA

Prayogo D, Cheng M-Y, Prayogo H (2017) A novel implementation of nature-inspired optimization for civil engineering: a comparative study of symbiotic organisms search. Civ Eng Dimens 19(1):36–43

Saremi S, Mirjalili S, Lewis A (2014) Biogeography-based optimisation with chaos. Neural Comput Appl 25(5):1077–1097

Sartakhti JS, Afrabandpey H, Saraee M (2017) Simulated annealing least squares twin support vector machine (SA-LSTSVM) for pattern classification. Soft Comput 21(15):4361–4373

Sousa JB, Craus J, Monismith C (1991) Summary report on permanent deformation in asphalt concrete. Strategic Highway Research Program (SHRP), National Research Council, University of California, Berkeley

Suykens JAK, Vandewalle J (1999) Least squares support vector machine classifiers. Neural Process Lett 9(3):293–300

Tran D-H, Cheng M-Y, Prayogo D (2016) A novel Multiple Objective Symbiotic Organisms Search (MOSOS) for time–cost–labor utilization tradeoff problem. Knowl Based Syst 94:132–145

van Gestel T, Suykens JAK, Baesens B, Viaene S, Vanthienen J, Dedene G, de Moor B, Vandewalle J (2004) Benchmarking least squares support vector machine classifiers. Mach Learn 54(1):5–32

Witczak MW, Kaloush KE, Pellinen T, El-Basyouny M, Von Quintus H (2002) Simple performance test for superpave mix design. NCHRP Report

Yan K-Z, Ge D-D, Zhang Z (2014) Support vector machine models for prediction of flow number of asphalt mixtures. Int J Pavement Res Technol 1(7):31–39

Yu VF, Redi AANP, Yang C-L, Ruskartina E, Santosa B (2017) Symbiotic organisms search and two solution representations for solving the capacitated vehicle routing problem. Appl Soft Comput 52:657–672

Zhenyu G, Bo C, Min Y, Binggang C (2006) Self-adaptive chaos differential evolution. In: Jiao L, Wang L, Gao X, Liu J, Wu F (eds) Advances in natural computation. Springer, Berlin, pp 972–975