Research Article

Optimization ELM Based on Rough Set for Predicting the Label of Military Simulation Data

Xiao-jian Ding and Ming Lei

Science and Technology on Information Systems Engineering Laboratory, Nanjing 210007, China

Correspondence should be addressed to Xiao-jian Ding; wjswsl@163.com

Received 19 April 2014; Revised 26 July 2014; Accepted 18 August 2014; Published 25 September 2014

Academic Editor: Yi Jin

Copyright © 2014 X.-j. Ding and M. Lei. 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.

By combining rough set theory with optimization extreme learning machine (OELM), a new hybrid machine learning technique is introduced for military simulation data classification in this study. First, multivariate discretization method is implemented to convert continuous military simulation data into discrete data. Then, rough set theory is employed to generate the simple rules and to remove irrelevant and redundant variables. Finally, OELM is compared with classical extreme learning machine (ELM) and support vector machine (SVM) to evaluate the performance of both original and reduced military simulation datasets. Experimental results demonstrate that, with the help of RS strategy, OELM can significantly improve the testing rate of military simulation data. Additionally, OELM is less sensitive to model parameters and can be modeled easily.

1. Introduction

Information warfare, which is the competitive use of information in modern forces, has recently become of increasing importance to the military domain. Information superiority and knowledge dominance are key points to winning the future war. There has been a substantial growth in the amount of stored information in military systems, mainly due to the rapid development of military information construction. However, the large amount of stored data becomes impracticable for specialists to analyze through conventional methods.

Military data research has gained great importance and interest in recent years, especially in operational areas. Data learning technology provides important analysis and decision services to aid the commander. To provide decision support for commanders, it is worth studying how to effectively learn the massive data that emerges in operations. The data used in the equipment war gaming system was classified on the basis of introducing terms about verification, validation, and certification (VV&C) and a sort criterion of authoritative data sources (ADS) was also put forward, as described by

Liguo et al. [1]. Li et al. [2] analyzed the characteristics of military data, put forward methods to classify data based on their characteristics and functions, and discussed data sources for combat simulation. Gao et al. [3] introduced a military simulation data aggregation framework for extraction, transformation, transportation, and loading based on message oriented middleware. As seen in [4], preliminary study on simulation data mining by rough set (RS) theory was done. RS enables us to find the dependencies and to reduce the number of attributes contained in military simulation data. With the help of RS, valuable results can be obtained by mining the battle simulation data, and the attributes which make a strong impact on performance of datasets learning can be found. However, given newly generated battle data, one cannot predict the classification results with the method of [4].

In this study, a new regularization algorithm-optimization extreme learning machine (OELM) [5], together with RS theory, is selected as main components for constructing an integrated predictor. OELM is an optimization algorithm based on conventional extreme learning machine (ELM) [6–11], which has the following

properties: first, compared to the traditional ELM, the minimization norm of output weights enables OELM to get better generalization performance; second, compared to support vector machine (SVM) [12], OELM finds the optimal solution in the search space of $[0, C]^N$, where SVM always searches the suboptimal solution of OELM due to equation constraints, so OELM usually achieves better generalization performance than a traditional SVM; third, OELM is less sensitive to specified parameters and can be implemented easily.

In this paper, the performance of an integrated method of multivariate discretization, RS theory, and OELM for military simulation data analysis is investigated and compared with state-of-the-art classifiers: ELM and SVM. As there exist some interactions between different attributes of military simulation data, multivariate discretization attempts to find the correct cuts by taking into account one attribute independently from the others. RS theory is adopted to reduce redundant attributes of military simulation data, and then OELM algorithm is employed to train this new dataset with optimal subset attributes.

We start this work with description of the integrated method in Section 2. Military simulation data preprocessing method is discussed in Section 3. Military simulation data analysis is shown in Section 4. Section 5 presents the experimental setup and details the results arrived at. Finally, Section 6 gives the conclusion.

2. Integrated Method

In this section, RS theory is adopted to reduce redundant attributes of the military simulation dataset, and then OELM algorithm is employed to train this new dataset with optimal attributes subset. The main contributions of this work are as follows: (1) multivariate discretization method is used to preprocess the original military dataset; (2) rules generated by RS method are used to create a reduced military dataset; (3) OELM, together with two other popular algorithms, is used to measure the generalization performance of both original dataset and reduced dataset.

2.1. OELM Review. For N arbitrary distinct samples $\{\mathbf{x}_i, t_i\}_{i=1}^N$, where $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{id}] \in \mathbf{R}^d$ and $t_i \in \{-1, 1\}$, the decision function given by an ELM is

$$f(\mathbf{x}) = \text{sign} \left(\sum_{i=1}^L \beta_i G(\mathbf{a}_i, b_i, \mathbf{x}) \right) = \text{sign}(\boldsymbol{\beta} \cdot h(\mathbf{x})), \quad (1)$$

where β_i is the output weight from the i th hidden node to the output node and $G(\mathbf{a}_i, b_i, \mathbf{x})$ is the output of the i th hidden node. $h(\mathbf{x}) = [G(\mathbf{a}_1, b_1, \mathbf{x}), \dots, G(\mathbf{a}_L, b_L, \mathbf{x})]$ is the output vector of the hidden layer, and it can map the training data $\mathbf{x}_i$ from the input space to the L -dimensional ELM feature space.

Based on the theory of Bartlett [13], feedforward neural network with smaller norm of weight, not only has smaller

training error, but also obtains better generalization performance. Therefore, ELM amounts to minimizing the training error with the minimized norm of the output weight:

$$\text{Minimize: } \sum_{i=1}^N \|\boldsymbol{\beta} \cdot h(\mathbf{x}_i) - t_i\| \quad (2)$$

$$\text{Minimize: } \|\boldsymbol{\beta}\|.$$

Any set of training data transformed from the input space to the ELM feature space with the mapping $h(\mathbf{x})$ is linearly separable. In order to prevent the classification problem of overfitting, variables ξ_i , $i = 1, \dots, N$ are introduced and one can minimize the testing error as follows:

$$\begin{aligned} \boldsymbol{\beta} \cdot h(\mathbf{x}_i) &\geq 1 - \xi_i \quad \text{for } t_i = +1, \\ \boldsymbol{\beta} \cdot h(\mathbf{x}_i) &\leq -1 + \xi_i \quad \text{for } t_i = -1. \end{aligned} \quad (3)$$

Two inequalities in (3) can be combined into one set of inequalities:

$$t_i \cdot \boldsymbol{\beta} \cdot h(\mathbf{x}_i) \geq 1 - \xi_i \quad \forall i. \quad (4)$$

Thus, OELM works based on solving the following quadratic program mathematical model:

$$\begin{aligned} \text{Minimize: } & \frac{1}{2} \|\boldsymbol{\beta}\|^2 + C \sum_{i=1}^N \xi_i \\ \text{Subject to: } & t_i \cdot \boldsymbol{\beta} \cdot h(\mathbf{x}_i) \geq 1 - \xi_i, \quad i = 1, \dots, N \\ & \xi_i \geq 0, \quad i = 1, \dots, N. \end{aligned} \quad (5)$$

Compared to SVM's optimization problem, there are three differences between SVM and OELM:

- (1) The mapping $\phi(\mathbf{x}_i)$ used in SVM is usually unknown and cannot be computed directly. The function $h(\mathbf{x}_i)$ used in OELM can be any bounded nonconstant piecewise continuous activation function for additive node, which can be calculated exactly.
- (2) Kernel parameters used in SVM need to be tuned manually, whereas all parameters of $h(\mathbf{x})$ for OELM are selected randomly.
- (3) The bias b is not needed in the constraints of OELM's optimization problem, because the separating hyperplane of OELM tends to pass through the origin.

2.2. Karush-Kuhn-Tucker Conditions. According to Lagrange optimization theory, minimizing the norm of output weights is equivalent to solving the dual optimization problem of (5):

$$\text{Minimize: } \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N t_i t_j K_{\text{ELM}}(\mathbf{x}_i, \mathbf{x}_j) \alpha_i \alpha_j - \sum_{i=1}^N \alpha_i \quad (6)$$

$$\text{Subject to: } 0 \leq \alpha_i \leq C, \quad i = 1, \dots, N.$$

ELM kernel matrix can be defined as

$$K_{\text{ELM}}(\mathbf{x}_i, \mathbf{x}_j) = h(\mathbf{x}_i) \cdot h(\mathbf{x}_j). \quad (7)$$

For the sake of convenience, (6) can be written compactly as

$$\begin{aligned} \text{Minimize: } f(\alpha) &= \frac{1}{2} \alpha^T K \alpha - \mathbf{e}^T \alpha \\ \text{Subject to: } 0 &\leq \alpha_i \leq C, \quad i = 1, \dots, N, \end{aligned} \quad (8)$$

where $f : R^N \rightarrow R$ is a convex function, $K = K_{\text{ELM}} \mathbf{t} \mathbf{t}^T \in \mathbf{R}^{N \times N}$ is positive semidefinite matrix, $\alpha = [\alpha_1, \dots, \alpha_N]^T \in R^N$, and $\mathbf{e} = [1, \dots, 1]^T \in R^N$.

Let $g(\alpha) = \partial f(\alpha) / \partial \alpha = K \alpha - \mathbf{e}$. Let $g_i(\alpha)$ denote the i th element of $g(\alpha)$; that is, $g_i(\alpha) = [K \alpha - \mathbf{e}]_i$. The Lagrange function of (8) is

$$L(\alpha, \eta, \rho) = f(\alpha) - \sum_{i=1}^N \eta_i \alpha_i - \sum_{i=1}^N \rho_i (C - \alpha_i), \quad (9)$$

where η_i and ρ_i are the Lagrange multipliers and are nonnegative values.

In order to find the optimal solutions of (9) we should have

$$\frac{\partial L(\alpha, \eta, \rho)}{\partial \alpha_i} = 0 \implies g_i(\alpha) - \eta_i + \rho_i = 0. \quad (10)$$

Based on the Karush-Kuhn-Tucker (KKT) theorem, the KKT conditions of (10) should be

primal feasibility

$$0 \leq \alpha_i \leq C, \quad \forall i; \quad (11)$$

dual feasibility

$$\eta_i \geq 0, \quad \forall i, \quad (12)$$

$$\rho_i \geq 0, \quad \forall i; \quad (13)$$

complementary slackness

$$\eta_i \alpha_i = 0, \quad \forall i, \quad (14)$$

$$\rho_i (C - \alpha_i) = 0, \quad \forall i. \quad (15)$$

Furthermore, consider the following.

- (1) If $\alpha_i = 0$, from (15), we have $\rho_i = 0$. Thus, from (10) and (12), we have

$$g_i(\alpha) \geq 0. \quad (16)$$

- (2) If $0 < \alpha_i < C$, from (14) and (15), we have $\eta_i = 0$ and $\rho_i = 0$. Thus, from (10), we have

$$g_i(\alpha) = 0. \quad (17)$$

- (3) If $\alpha_i = C$, from (14), we have $\eta_i = 0$. Thus, from (10) and (13), we have

$$g_i(\alpha) \leq 0. \quad (18)$$

The KKT conditions are both necessary and sufficient for optimality. Thus, (8) is solved when for all i

$$\begin{aligned} \alpha_i = 0 &\iff g_i(\alpha) \geq 0, \\ 0 < \alpha_i < C &\iff g_i(\alpha) = 0, \\ \alpha_i = C &\iff g_i(\alpha) \leq 0. \end{aligned} \quad (19)$$

Active set approach is the best choice to solve (8) because it is an iterative algorithm and it maintains feasibility of the vector α (i.e., $0 \leq \alpha_i \leq C$) while iteration continues until KKT conditions (19) are satisfied.

2.3. Rough Set. Rough set (RS) term, which was first introduced by Pawlak et al. [14] has become a popular pattern recognition tool for generating logical rules for prediction. Attribute reduction, which is one of the core parts of RS theory, means to find out the most informative attributes and remove the irrelevant or unimportant attributes with minimal information loss.

RS is the approximation of an uncertain set by a pair of precise concepts called lower and upper approximations. The lower approximation comprises elements belonging to it, whereas the upper approximation of the set includes elements which are possible members of the set.

RS model is defined in terms of an information system, which can be defined formally as 4-tuple in [15, 16]:

$$S = \langle U, Q, V, f \rangle, \quad (20)$$

where $U = \{u_1, u_2, \dots, u_n\}$ is a nonempty set of objects called the universe, $Q = \{q_1, q_2, \dots, q_n\}$ is a finite set of attributes, V_q is the domain of attribute q , and $V = \bigcup_{q \in Q} V_q$. $f : U \times Q \rightarrow V$ is an information function that $f(x, q) \in V_q$ for each $q \in Q$, $x \in U$. Each object of universe is described by a vector

$$\text{Des}_q(x) = [f(x, q_1), f(x, q_2), \dots, f(x, q_m)], \quad (21)$$

where $x \in U$. To every nonempty subset of attributes P is associated with an indiscernibility relation on U , denoted by I_P :

$$I_P = \{(x, y) \in U \times U : f(x, q) = f(y, q), \forall q \in P\}. \quad (22)$$

Equation (22) is an equivalence relation. The family of all the equivalence classes of the I_P is denoted by U / I_P and class containing an element x by $I_P(x)$.

Formally, let X be a nonempty set of U . Set X is approximated by means of P -lower and P -upper approximations of X :

$$\begin{aligned} \underline{P}(X) &= \{x \in U : I_P(x) \subseteq X\}, \\ \overline{P}(X) &= \bigcup_{x \in X} I_P(x). \end{aligned} \quad (23)$$

The P -boundary of X is denoted by $B_n(X)$

$$B_n(X) = \overline{P}(X) - \underline{P}(X). \quad (24)$$

The following relation holds: $\underline{P}(X) \subseteq X \subseteq \overline{P}(X)$.

There are three regions of interest, they are the inner region for objects in $\underline{P}(X)$, the outer region for objects not in $\overline{P}(X)$, and the boundary region for the objects that are uncertain.

TABLE 1: Parameters of simulation data.

	W1	T1	S	V	W2	T2	F	P	R
1	Fine	Flat	Fast	More	Small	Short	Poor	Normal	High
2	Fine	Undulating	Slow	More	Small	Short	Poor	Normal	Low
3	Fog	Flat	Slow	Less	Small	Long	Better	Normal	High
4	Fog	Flat	Fast	Less	Big	Long	Better	Nervous	Low
5	Fine	Undulating	Slow	Less	Big	Long	Better	Normal	High
6	Fine	Undulating	Fast	Less	Big	Short	Poor	Nervous	Low
7	Fine	Flat	Slow	More	Small	Short	Better	Nervous	High
8	Fog	Flat	Slow	More	Small	Short	Poor	Nervous	Low
9	Fine	Undulating	Fast	Less	Big	Short	Better	Nervous	High
10	Fog	Undulating	Fast	Less	Small	Long	Better	Normal	Low
11	Fine	Flat	Fast	Less	Small	Short	Better	Normal	High
12	Fog	Flat	Fast	More	Big	Short	Poor	Normal	Low

TABLE 2: Converted attribute of “Weather.”

Attribute	Fine	Fog
Weather = fine	1	0
Weather = fog	0	1

3. Military Simulation Data Preprocessing

The performance of the integrated method is evaluated on countermeasure simulation data of tank formation [4], which was exercised on medium undulating ground, hills, and Taiwan highland for digital armored battalion simulation system. The specification of the dataset is presented in Table 1.

From Table 1, items “W1,” “T1,” ..., and “R” represent “Weather,” “Terrain,” “Speed,” “Vegetation,” “Wind,” “Time,” “Formation,” “Psychology,” and “Shooting rate,” respectively. ELM algorithm can be applied only to datasets composed of categorical attributes but attributes in Table 1 are continuous variables. Discretization process is important for char variables because it is less prone to variance in estimation from small fragmented data and it provides better performance for rule extraction. The main discretization process methods are supervised [17], hierarchical [18], top-down [19], and multivariate [20]. The last one is employed in our work.

Multivariate discretization quantifies simultaneously many features. In the preprocessing phase, attribute “Weather” is converted into two new attributes “fine” and “fog,” which is shown in Table 2.

Based on examples shown in Table 2, other attributes are converted in Table 3.

4. Military Simulation Data Analysis

Generally speaking, it is a data mining task to distinguish the probability of shooting rate. The goal of military data analysis is to predict the unknown value of shooting rate, such as high or low. More specifically, military simulation data analysis process can be modeled as a classification problem.

A typical military dataset is sparse compared to a traditional classification dataset. It usually contains small samples but with many features. This kind of sparseness often

FIGURE 1: Military simulation data visualization.

descends the performance of some classifiers. To improve the classification rate, one can remove the irrelevant or redundant features. Intuitively speaking, a classifier trained in lower-dimensional feature space is expected to capture the inherent data distribution and performs better in such feature space.

There are many advantages of variable and feature selection, such as facilitating data visualization and data understanding, reducing training and testing time, and so forth. We give a 3-dimensional visualization of dataset discussed in Section 4, as seen in Figure 1.

In Figure 1, only 4 samples can be seen. As there are 12 samples for entire dataset, we conjecture that some samples overlap each other. It can be obviously concluded that much redundant information is contained in this dataset.

RS method is a well-established method for feature selection. It is widely used in many applications and more flexible than other methods, such as principal component analysis (PCA) [21]. One can use RS method to discover the data dependencies and reduce the number of attributes. For comparison purpose we used five other widely used feature selection methods (i) PCA, (ii) multidimensional scaling (MDS), (iii) generalized discriminant analysis (GDA), (iv) factor analysis (FA), and (v) Isomap.

TABLE 3: Discretization dataset.

W1		T1		S		V		W2		T2		F		P		R
1	0	1	0	1	0	1	0	1	0	0	1	0	1	1	0	1
1	0	0	1	0	1	1	0	1	0	0	1	0	1	1	0	0
0	1	1	0	0	1	0	1	1	0	1	0	1	0	1	0	1
0	1	1	0	1	0	0	1	0	1	1	0	1	0	0	1	0
1	0	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1
1	0	0	1	1	0	0	1	0	1	0	1	0	1	0	1	0
1	0	1	0	0	1	1	0	1	0	0	1	1	0	0	1	1
0	1	1	0	0	1	1	0	1	0	0	1	0	1	0	1	0
1	0	0	1	1	0	0	1	0	1	0	1	1	0	0	1	1
0	1	0	1	1	0	0	1	1	0	1	0	1	0	1	0	0
1	0	1	0	1	0	0	1	1	0	0	1	1	0	1	0	1
0	1	1	0	1	0	1	0	0	1	0	1	0	1	1	0	0

TABLE 4: Reduced military simulation dataset.

W1		T1		S		F		R
1	0	1	0	1	0	0	1	1
0	1	1	0	0	1	1	0	1
1	0	0	1	0	1	1	0	1
1	0	1	0	0	1	1	0	1
1	0	0	1	1	0	1	0	1
1	0	1	0	1	0	1	0	1
1	0	0	1	0	1	0	1	0
0	1	1	0	1	0	1	0	0
1	0	0	1	1	0	0	1	0
0	1	1	0	0	1	0	1	0
0	1	0	1	1	0	1	0	0
0	1	1	0	1	0	0	1	0

After feature selection, more or less overlapping phenomenon appears in Figure 2, besides the ‘Rough set’ subfigure. The main goal of feature selection method is to choose a subset of useful variables, which exclude many redundant, but relevant variables. As seen in Figure 2, RS method obtains the most relevant variables, because ‘Rough set’ subfigure has the most overlapping region. The usefulness of these variables is directly judged by the estimated accuracy of the learning method. Learning method based on features chosen with RS method achieves error rates lower than other methods.

5. Performance Evaluation

In this section we report the performance of the proposed integrated method both on original and reduced military simulation datasets and compare the performance with other state-of-the-art classifiers. The reduced dataset is preposessed using the procedure described in Section 4. After converting, 8 attributes in the original data are scattered to 16 attributes. In our classification application, the inputs (attributes) are normalized into the range $[0, 1]$, while the outputs (targets) are normalized into the range $[-1, 1]$. All the

simulations are running in the MATLAB R2008a (Windows version) environments with Intel 3.0 GHZ and 2 G RAM.

For comparison purpose we used two other widely used classification methods, ELM and SVM. For the implementation of SVM, we use the Spider library for MATLAB, which is publicly available from [22]. Both MATLAB codes of ELM and OELM can be downloaded from ELM host site: <http://www.ntu.edu.sg/home/egbhuang/>. To train SVM, radial basis function (RBF) kernel $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$ is used in our experiments. ELM with sigmoidal activation function $G(a, b, x) = 1/(1 + \exp(-(a \cdot x + b)))$ is tested. Both the Gaussian RBF activation function $G(a, b, x) = \exp(-\|x - a\|^2/b)$ and the sigmoidal additive activation function have been used in the simulation of OELM. For both ELM and OELM with sigmoidal additive activation function and RBF activation function, the input weights a and biases b are randomly generated from $(-1, 1)^N \times (0, 1)$ based on the uniform probability distribution.

5.1. Reduced Dataset. According to the RS strategy in [4], the number of attributes is reduced from 16 to 8 on military simulation dataset, which is shown in Table 4.

5.2. Selection of Parameters. Two parameters for RBF kernel of SVM, the cost constant C and the kernel parameter γ , are both sensitive to the generalization performance. A straightforward way for model selection is the grid search strategy [23]. According to the method of [24], two parameters are tuned on a grid space of $\{10^{-3}, 10^{-2}, 0.05, 10^{-1}, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 10^2, 10^3, 10^4\} \times \{10^{-3}, 10^{-2}, 10^{-1}, 0.2, 0.4, 0.8, 1, 2, 5, 10, 20, 50, 10^2, 10^3, 10^4\}$. Therefore, for each problem we try $15 \times 15 = 225$ combinations of parameters (C, γ) for SVM. Figure 3 shows the generalization performance of SVM for each combination.

As can be seen in Figure 3, the specific C and kernel parameter γ value form a grid in the (C, γ) parameter space. It is clear that the wider the search space is, the more possibility the grid search method finds the best parameter combination. However, the grid search method exhaustively searches the grid to find a best combination. Figure 3(a)

FIGURE 2: Comparison of 6 feature selection methods.

FIGURE 3: The performance of SVM for 225 combinations.

TABLE 5: Comparison between three ELM, SVM, and OELM for two datasets.

Datasets	Classifiers	$L/(L, C)/(C, \gamma)$	Training time (s)	TR (%)	Dev (%)
Original dataset	ELM	5	$2.6243e - 004$	55.00	3.85
	SVM	$(1, 10^{-2})$	0.0079	59.33	2.67
	OELM-S	$(5, 10^2)$	0.0013	65.12	2.89
	OELM-R	$(5, 10^2)$	$8.5874e - 004$	63.87	2.95
Reduced dataset	ELM	5	$2.7404e - 004$	60.33	2.79
	SVM	$(1, 10^{-2})$	0.0077	60.00	1.85
	OELM-S	$(5, 10^3)$	0.0018	74.67	2.06
	OELM-R	$(6, 10^2)$	$9.8123e - 004$	72.92	2.36

TABLE 6: Random permutation performance comparison of ELM, SVM, and OELM-S.

Trials	Random permutation (training set)	TR (%)					
		Original dataset			Reduced dataset		
		ELM	SVM	OELM-S	ELM	SVM	OELM-S
1	(1, 8, 10, 2, 3, 12)	50.00	33.33	66.67	66.67	33.33	83.33
2	(3, 11, 6, 9, 10, 5)	50.00	33.33	50.00	50.00	33.33	66.67
3	(3, 9, 6, 11, 8, 10)	83.33	100	83.33	83.33	100	100
4	(8, 3, 11, 1, 10, 5)	50.00	33.33	50.00	66.67	33.33	83.33
5	(9, 6, 3, 1, 8, 4)	50.00	100	66.67	50.00	100	83.33
6	(9, 8, 12, 7, 1, 6)	50.00	100	66.67	66.67	100	83.33
7	(5, 9, 7, 10, 3, 6)	50.00	33.33	33.33	66.67	33.33	66.67
8	(5, 2, 4, 10, 6, 8)	33.33	16.67	16.67	33.33	16.67	50.00
9	(3, 5, 11, 8, 10, 12)	66.67	100	66.67	83.33	100	100
10	(5, 2, 7, 12, 9, 1)	50.00	33.33	50.00	50.00	33.33	66.67

indicates that the best generalization performance is achieved for 18 combinations on original dataset, and the performance of SVM highly depends on the combination of parameters. Similar results are obtained in Figure 3(b).

For ELM, there is only one parameter L (optimal number of hidden nodes) that needs to be determined. During our simulation it is found that ELM with 5 hidden nodes can generally obtain good performance for both datasets. Generally speaking, the generalization performance of OELM is not sensitive to the number of hidden nodes L [5]. We set the number of hidden nodes 1, 2, 3, 4, 5, and 6. Moreover, there is one more parameter that needs to be determined, which is the number of cost parameter C . Similar to [5], we set this parameter 10^{-3} , 10^{-2} , 0.05 , 10^{-1} , 0.2 , 0.5 , 1 , 2 , 5 , 10 , 20 , 50 , 10^2 , 10^3 , and 10^4 . The optimal parameter(s) L (L, C) are selected to obtain the best testing performance. Table 5 shows the optimal parameters of three classifiers on two datasets.

5.3. Performance Comparison on Two Datasets. We evaluate the performance of three classifiers by repeated random splitting; that is, two datasets are equally partitioned into one training set and one test set. To avoid selection bias, there are at least one positive sample (labeled as “1”) and one negative sample (labeled as “-1”) in the training set. Such splitting is repeated 50 times. Training time, testing rate (TR) averaged over the 50 trials, and the corresponding standard deviations (Dev) are reported.

Table 5 draws a comparison between three classifiers, as well as some comparative results on different kernels. All of results represent the average, taken over 50 trials. OELM-S and OELM-R mean OELM using Sigmoidal and RBF kernel, respectively. As observed from Table 5, the performance of OELM-S and that of OELM-R is similar to each other except that OELM-S requires twice training time taken by OELM-R. We can see that the training time taken by ELM is much less than other classifiers, and the worst performance is also obtained by ELM. The performance of OELM-S or OELM-R is obviously higher than other classifiers. In order to further study this phenomenon, an extra experiment is conducted in Table 6.

Seen from Table 6, among 10 trials, the number of training and test set is fixed according to the above setting, but the order of training set is randomly shuffled for each trial. Although 100% testing rate is obtained for SVM on 4 trials, testing rate on other trials is no more than 33.33%. These results seem very unstable due to the randomness. Interestingly, OELM-S runs more stable than the other two classifiers, and this is why the performance of OELM-S is much higher than SVM and ELM.

Strictly speaking, it seems that, after data reduction, the performance of SVM cannot be improved on all trials. Moreover, OELM-S is exactly a good choice employed in our integrated strategy for improving the generalization performance of military simulation data.

6. Conclusions

This paper has introduced an integrated learning method of RS and OELM for military simulation dataset. The generalization performance of OELM is less sensitive to the user specified parameters especially for the number of hidden nodes. Compared to SVM, one can use OELM easily and effectively without parameter(s) tuning process. Significantly better results for the reduced dataset are obtained by employing OELM classifier over crisp discretization. This method has the potential for further military application because it can provide real-time decision support for commanders in battlefield.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] W. Ligu, X. Qing, and M. Xianquan, "The research of the data VV&C for the equipment war gaming based on the HLA," *Computer Engineering and Applications*, vol. 21, pp. 200–202, 2006.
- [2] M.-Z. Li, J.-K. Zhang, and W.-F. Che, "Research on data for combat simulation," *Command Control & Simulation*, vol. 32, no. 4, pp. 71–74, 2010.
- [3] H. Gao, H. Zhang, G. Chen et al., "Research and implementation of military simulation," *Fire Control & Command Control*, vol. 34, no. 2, pp. 150–153, 2009.
- [4] W.-M. Zhang and Q. Xue, "Application of rough set in date mining of warfare simulation," *Journal of System Simulation*, vol. 18, no. 2, pp. 179–181, 2006.
- [5] G.-B. Huang, X. Ding, and H. Zhou, "Optimization method based extreme learning machine for classification," *Neurocomputing*, vol. 74, no. 1–3, pp. 155–163, 2010.
- [6] G.-B. Huang and C.-K. Siew, "Extreme learning machine: RBF network case," in *Proceedings of the International Conference on Control, Automation, Robotics and Vision*, pp. 1651–1663, 2004.
- [7] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.
- [8] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.
- [9] J. W. Cao, T. Chen, and J. Fan, "Fast online learning algorithm for landmark recognition based on BoW framework," in *Proceedings of the 9th IEEE Conference on Industrial Electronics and Applications*, Hangzhou, China, June 2014.
- [10] J. W. Cao and L. Xiong, "Protein sequence classification with improved extreme learning machine algorithms," *BioMed Research International*, vol. 2014, Article ID 103054, 12 pages, 2014.
- [11] J. W. Cao, Z. Lin, G.-B. Huang, and N. Liu, "Voting based extreme learning machine," *Information Sciences*, vol. 185, pp. 66–77, 2012.
- [12] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [13] P. L. Bartlett, "The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 525–536, 1998.
- [14] Z. Pawlak, J. Grzymala-Busse, R. Slowinski, and W. Ziarko, "Rough sets," *Communications of the ACM*, vol. 38, no. 11, pp. 88–95, 1995.
- [15] Z. Pawlak and A. Skowron, "Rudiments of rough sets," *Information Sciences*, vol. 177, no. 1, pp. 3–27, 2007.
- [16] S. Greco, M. Benedetto, and R. Slowinski, "New developments in the rough set approach to multi-attribute decision analysis," *Bulletin of International Rough Set Society*, vol. 2, no. 2–3, pp. 57–87, 1998.
- [17] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and unsupervised discretization of continuous features," in *Proceedings of the 12th International Conference on Machine Learning*, pp. 194–202, 1995.
- [18] R. Kerber, "Discretization of numeric attributes," in *Proceedings of the 10th National Conference on Artificial Intelligence*, pp. 123–128, MIT Press, Cambridge, Mass, USA, 1992.
- [19] F. Hussain, H. Liu, C. L. Tan, and M. Dash, "Discretization: an enabling technique," Tech. Rep., School of Computing, Singapore, 1999.
- [20] S. D. Bay, "Multivariate discretization of continuous variables for set mining," in *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 315–319, August 2000.
- [21] T. C. Lei, S. Wan, and T. Y. Chou, "The comparison of PCA and discrete rough set for feature extraction of remote sensing image classification—a case study on rice classification, Taiwan," *Computational Geosciences*, vol. 12, no. 1, pp. 1–14, 2008.
- [22] The Spider Library for MATLAB, <http://www.kyb.tuebingen.mpg.de/bs/people/spider/>.
- [23] N. Ancona, C. Cicirelli, E. Stella, and A. Distanto, "Object detection in images: run-time complexity and parameter selection of support vector machines," in *Proceedings of the 16th International Conference on Pattern Recognition*, vol. 2, pp. 426–429, August 2002.
- [24] P. Ghanty, S. Paul, and N. R. Pal, "NEUROSVM: an architecture to reduce the effect of the choice of kernel on the performance of SVM," *Journal of Machine Learning Research*, vol. 10, no. 3, pp. 591–622, 2009.