Sliding Mode Control based Support Vector Machine RBF Kernel Parameter Optimization

Maryam Yalsavar
School of Electrical and Computer Engineering
Shiraz University
Shiraz, Iran
m.yalsavar@shirazu.ac.ir

Paknoosh Karimaghaee
School of Electrical and Computer Engineering
Shiraz University
Shiraz, Iran
kaghaee@shirazu.ac.ir

Akbar Sheikh-Akbari
School of Built Environment, Engineering and Computing
Leeds Beckett University
Leeds, UK
a.sheikh-akbari@leedsbeckett.ac.uk

Jamshid Dehmeshki
School School of Computer Science and Mathematics
Kingston University
London, UK
j.dehmeshki@kingston.ac.uk

Mohammad-Hassan Khooban
Department of Engineering - Applied Formal Methods
Aarhus University
Aarhus, Denmark
khooban@eng.au.dk

Salah Al-Majeed
School of Business and Technology
University of Gloucestershire
Cheltenham, UK
SalahAlMajeed @ieee.org

***Abstract*— Support Vector Machine (SVM) is a learning-based algorithm, which is widely used for classification in many applications. Despite its advantages, its application to large scale datasets is limited due to its use of large number of support vectors and dependency of its performance on its kernel parameter. This paper presents a Sliding Mode Control based Support Vector Machine Radial Basis Function's kernel parameter optimization (SMC-SVM-RBF) method, inspired by sliding mode closed loop control theory, which has demonstrated significantly higher performance to that of the standard closed loop control technique. The proposed method first defines an error equation and a sliding surface and then iteratively updates the RBF's kernel parameter based on the sliding mode control theory, forcing SVM training error to converge below a predefined threshold value. The closed loop nature of the proposed algorithm increases the robustness of the technique to uncertainty and improves its convergence speed. Experimental results were generated using nine standard benchmark datasets covering wide range of applications. Results show the proposed SMC-SVM-RBF method is significantly faster than those of classical SVM based techniques. Moreover, it generates more accurate results than most of the state of the art SVM based methods.**

Keywords—support vector machine, sliding mode control, radial basis function, classification, classification speed.

I. INTRODUCTION

Support Vector Machine (SVM) is a machine learning algorithm that widely used for classification. SVM is one of the robust and efficient classification methods amongst the well know classification algorithms such as nearest neighbor, boosted decision trees, regularized logistic regression, neural networks, and random forests [1], [2], [3]. When dealing with non-linearly separable data, SVM maps the data into higher dimensional space using kernels prior to performing the classification [4]. SVM formulates a quadratic programming (QP) problem to find a separating hyperplane, which maximizes the margin between two classes of the data [3], [5], [6]. Since SVM achieves a unique solution and learns from dimensionality of feature space, it is more robust than other techniques to over fitting [4], [6], [7]. Despite all the advantages and applications of the SVM [8], [9], its classification speed is deteriorated when dealing with large scale problems as it uses large number of support vectors. In addition, its training computationally expensive and timely [10], [11]. Over the last two decades, many techniques have been proposed to speed up the test and training time of the SVM [5], [10], [11], [12], [13], [14], [15], [16], [17], [18] which have been resulted in techniques that reduce the number of SVs. However, there are demands for more powerful techniques. In some branches of control such as nonlinear [19], [20] and optimal control [21] SVM has been used due to its capabilities. However, the application of the Sliding Mode Control (SMC) to speed up the training period of the SVM and improving its performance has not been reported in the literature.

This paper presents a closed loop method based on Sliding Mode Control (SMC) to find optimum value for RBF kernel parameter of the SVM. Application of the slide mode control significantly improves the learning speed of the SVM and improves the performance of the resulting classifier in term of accuracy and matching computation cost. The proposed method uses a first order equation solution to solve an n-order problem for non-linear systems with some level of uncertainty. The proposed method first defines an error equation and the sliding surface and then uses the closed loop SMC algorithm to find an optimum value for RBF kernel parameter, γ, of the SVM. This significantly reduces the SVM training time. Experimental results on nine benchmark datasets show that the proposed method significantly outperforms the anchor SVM in terms of accuracy at lower number of Support Vectors (SV). This implies that the proposed method is faster than the anchor SVM. Experimental results also show that the proposed method achieves superior or very competitive performance in term of accuracy to those of the state-of-the-art techniques. Results also shows that the propose method is faster than the state-of-the-art techniques in term of classification time, as it generates smaller number of SV. The rest of this paper is organized as follows. Sections II and III give an overview on Support Vector Machine (SVM) and the Sliding Mode Control (SMC) technique, respectively. The proposed method is introduced in Section IV. Section V presents the experimental results and finally paper is concluded in Section VI.

II. SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is a linear classifier, which is widely used to split a linearly separable dataset into its two classes. It determines an optimum hyperplane to classify the input dataset that maximizes its distance and margin from the data of the both classes. Hard margin SVM is usually used to classify data within a linearly separable dataset. Assume that there are n data points with labels either 1 or -1 are in the input dataset. SVM takes the following steps to find the initial margin and then optimize it. Assume that $w^T x + b = 0$ is the hyperplane equation, where w is an orthogonal vector to the hyperplane and b is its bias, then the distance of a point to the hyperplane can be determined using (1):

$$d_i(x) = \frac{w^T x_i + b}{\|w\|} \quad (1)$$

where x_i is a data point and $d_i(x)$ is x_i's signed distance from the hyperplane, the sign of the $d_i(x)$ is x_i's label and shown by y_i. The margin is the defined by $min\left\{y_i \frac{w^T x_i + b}{\|w\|}\right\}$. To make the distance of all points to the hyperplane greater than 1, w and b can be rescaled as follows:

$$y_i(w^T x_i + b) \geq 1 \xrightarrow{yields} margin = \frac{1}{\|w\|} \quad (2)$$

Therefore, the SVM algorithm searches for the maximum margin. Based on (2), this can be formulated as a Quadratic Problem (QP) as shown in equation 3:

$$\min_w \frac{1}{2}\|w\|^2$$
$$s.t.\ y_i(w^T x_i + b) \geq 1 \quad \forall i = 1, \dots, n \quad (3)$$

The resulting QP is a convex problem, which results in a global minimum or maximum solution. Consequently, the hyperplane w is calculated by solving this QP. Classifying a nonlinear dataset using a linear algorithm such as SVM can be achieved by reshaping and increasing the dimension of the data. However, this results in the curse of dimensionality. To overcome this issue, SVM uses the concept of the kernel, which is known as soft margin SVM. In this case, the decision boundary is non-linear, and the data is not linearly separable. Hence, some points within the dataset may cross the margin or not correctly classified. Therefore, the hard margin SVM's constrain is not valid anymore. The constrain can be modified to include the nonlinear cases as well, as shown in (4), [22]:

$$\min_w \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i$$
$$s.t.\ y_i(w^T x_i + b) \geq 1 - \xi_i,\ where\ \xi_i \geq 0 \quad (4)$$

In equation 4, ξ_i is added to the constrain for the points, which violate the constrain. But by changing the constraint in this way, all points within the dataset can violate the constraint. Therefore, the number of points, which can violate the margin are restricted by adding a penalty or regularization parameter, C. One can solve the dual formula (4) as [23], [24], [25]:

$$\max_{\alpha_i} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j {x_i}^T x_j$$
$$s.t. \textstyle\sum_{i=1}^{n} \alpha_i y_i = 0\ ,\quad 0 \leq \alpha_i \leq C \quad (5)$$

Where α_i is a dual variable, which is obtained via the QP. The points with α_i greater than zero are support vectors and the points with α_i equal to C are the ones that violate the constrain of the hard margin SVM. Finally, the class of each input data point by using RBF kernel can be determined as follows: $y = sign[\sum_{i=1}^{n} \alpha_i y_i exp(-\gamma\|x - x_i\|^2) + b]$, where n is the number of training data, α_i is the dual variable, x_i is the training data, x is the input data point, y_i is the corresponding label of x_i and γ is RBF kernel parameter.

III. SLIDING MODE CONTROL

Sliding Mode Control (SMC) is a powerful technique for controlling a non-linear system, particularly when there is not a precise mathematical model for the system or the model does not represent all system's parameters [26], [27], [28]. SMC assumes that controlling a first order system is much easier than controlling an nth-order system. This allows an nth-order problem to be replaced by its equivalent first order problem. For the transformed problem, perfect performance can be achieved in principle despite the presence of inaccuracy of arbitrary parameters [26]. This creates a sliding surface and drives the state of the system toward the surface in its state space. Once the state of the system reached the sliding surface, SMC keeps the state of the system in a close neighborhood of the sliding surface [28]. SMC consists of two parts: the sliding surface and the off-surface dynamics. For a single input dynamic system of form $x = f(x) + b(x)u$,where x is the scaler output, u is the scaler input, $\boldsymbol{x}$ is the state vector, $f(x)$ and $b(x)$ are system model, which are not exactly known and have uncertainties. The track error can be written as:

$$e = x - x_d \quad (6)$$

where x and x_d are the output and desired output respectively. A time varying surface S(t) in the state space R can be defined by the scaler space s(x; t)=0, where:

$$s(x;t) = (\frac{d}{dx} + \lambda)^{n-1} e \quad (7)$$

Where λ is a strictly positive constant and for n =2, (7) can be written as: $s = \dot{e} + \lambda e$.
The problem of tracking $x \equiv x_d$ is equivalent to that of remaining on the surface S(t) for all $t > 0$; indeed $s(x, t) \equiv 0$ represents a linear differential equation whose unique solution is $e \equiv 0$, given its initial condition. Thus, the problem of tracking the n-dimensional vector x_d can be reduced to that of keeping the scalar quantity s at zero.

$$s = \dot{e} + \lambda e \equiv 0 \quad (8)$$

When the surface is driven to zero, the error drives to zero too, for $t \to \infty$ [26]. To show that, we work backwards by postulating the off-surface dynamics that must be of the form:

$$\dot{S} = -f(S) \quad (9)$$

where f(S) could be any non-decreasing odd function. This shows that the change in S and the 'distance' of the current state of the sliding surface, it is always opposite the sign of S. The control input should force the states to approach it. So, $\dot{S}$ must be a function of the input, u. $\dot{S}$ must be a function of the second derivative of the error to just be a function of the input. This implies that S should only be a function of error and its first derivative. The simplest form of such function, which guarantees e→0 as t→∞ is given in equation 8 [28]. Consequently, as S approaches zero, so does the tracking error. For equation 8, the sliding surface is a line with the slope of $-\lambda$ in phase plane. Starting with any initial condition, the state trajectory drives to the sliding surface and then slides along the surface exponentially towards the desired value, x_d, with the time constant of $\frac{1}{\lambda}$, as This illustrated in Fig. 1 [26].
SVM has widely used to classify non-linear separable data where there is always some uncertainty in selection of its parameters such as regularization and kernel. This has inspired the author to use the concept of sliding mode control to improve the performance of the SVM algorithm.

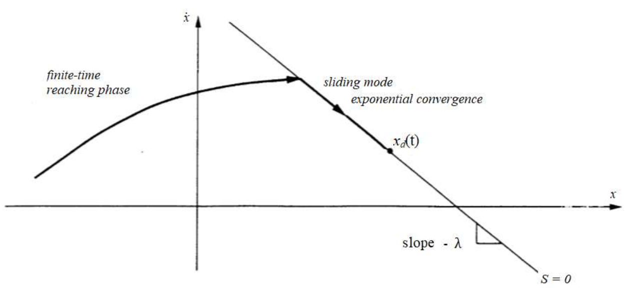

Fig.1. Graphical configuration of equation 8 [26].

IV. PROPOSED METHOD

Fig. 2 shows a block diagram of the proposed Sliding Mode Control based Support Vector Machine Radial Basis Function's kernel parameter optimization (SMC-SVM-RBF) method. The proposed method split the dataset into three parts named training, validation and test-subsets. The Trian SVM block takes the training subset and initial parameters including RBF kernel parameter, γ_{new} regularization parameters, C, λ, d, VE, which represent the state of the training error and train the SVM generating Support Vectors (SVs) and their numbers, N_{SVs}. The Classification block takes the SVs, N_{SVs} and the train subset, and classifies the train subset data into two classes. The classified train data are then assessed by the Assess classified data. If the Training Error (TE) equals to zero, it implies that the value of the RBF kernel parameter is not appropriate, and the optimization algorithm has arrived into a local minimum. Therefore, the value of the RBF kernel parameter is perturbed and backs to Training SVM block. This procedure is repeated until TE reaches a non-zero value. Classification and assessing the train and validation data generate the following information: the number of misclassified training data (MC), the labels of the misclassifiedtraining data (MC-lbs), the training error (TE) of the classified training data and the validation error (VE) of the classified validation data. MC and MC-lbs are used to update the RBF kernel parameter, TE is used to define the when it is necessary to perturb the RBF kernel parameter and VE is used as a measure to terminate the training procedure. For perturbing value of γ the algorithm checks, if TE is zero, γ_{old} will be perturbed until a non-zero TE is achieved. To perturb the value of the RBF kernel parameter, the value of the RBF kernel parameter is assessed when TE is zero; if its value is smaller than a threshold, it will be perturbed by a small value, otherwise it will be perturbed by a bigger value. The perturbing procedure can both increase or decrease the RBF kernel parameter. Experimental results presented in this paper are generated using small initial value for the RBF kernel parameter. When TE reaches a non-zero value, the training procedure starts as follows. First, three counters and three thresholds are initialized as follows: r1, r2, and r3 are set to one, thr1, thr2 and Maximum Number of iterations that are acceptable for enhancement in Validation Error (MNVE) are set to the Number of Mis-Classified train data (NMC), Number of Training Data (NTD), and a constant value, respectively. Then the algorithm goes through each element of Mis-Classified training data using its label, MC-lbs[r1], calculating its respected p and Q. If MC-lbs[r1] = -1, q will be calculated using $q = -\frac{1}{2} Q^{\dagger} p^T$. After that the algorithm goes through elements of q using counter r2 and for each positive element of q , ${\gamma_2}^{r2}$ is calculated, when all elements of ${\gamma_2}^{r2}$ are calculated, it computes $\gamma_1 = \frac{1}{l} \sum_{i=1}^{l} \gamma_2^i$ but if MC-lbs[r1] in not equal to -1, it assigns γ_{new} to γ_1 . The algorithm then assigns $\gamma_1$and 0 to γ' and γ_1 , respectively and increment r1 to point to the next mis-classified train data. This procedure is repeated for all mis-classified train data (NMC). When γ' is calculated for all misclassified train data, the algorithm will check r3, to see if r3 has reached its Maximum Number of iterations that are acceptable for enhancement in Validation Error (MNVE) threshold value. If not, a new value for γ is calculated as $\gamma_{new} = \sum_{j=1}^{m} \gamma'$ and it backs to Train SVM block and the procedure is repeated until MNVE reaches its predefined threshold, otherwise the training is completed and γ_{new} is taken γ and used to calculate the SVs. The resulting SVs are used to classify the test subset. The proposed Sliding Mode Control based Support Vector Machine Radial Basis Function's kernel parameter optimization (SMC-SVM-RBF) method has been mathematically proved but the proof has not been included in this manuscript.

V. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed SMC-SVM-RBF method, experimental results were generated using nine datasets from UCI machine learning repository [29] called: Letter Recognition (LR) (letters 'A' and 'N' are used for this experiment), Wisconsin Breast Cancer (WBC), Liver Disorder (LD), Heberman, Diabetes, Heart Disease, Ionosphere and Sonar datasets. The number of instances and dimension of the datasets used in this experiment, are tabulated in Table I.

To generate experimental results, all the datasets are normalized and then each dataset is randomly divided into three subsets called: train, test and validation subsets of size 70, 20 and 10 percent, respectively. The following setting are used to generate results: $f(S) = 50 * \arctan(S/10)$, $\lambda = 0.3$ and regularization parameter, C = 100.1. The results are presented in two sections. In section 1, the number of resulting numbers of Support Vectors (SVs), achieved accuracy for the train and test data of the proposed technique are compared to those of anchor SVM and tabulated in Table II. From Table II, the proposed technique generates significantly higher performance in term of accuracy and number of SVs than that of anchor SVM. In section 2, the performance of the proposed methods is compared to those of Zhang's and Zhiliang Liu's methods [16], [12], using four datasets and the results are tabulated in Table III. From Table III, it can be seen that the propose method gives either superior or very competitive results to those of Zhang's and Zhiliang Liu's methods.

TABLE I NUMBER OF INSTANCES AND DIMENSION OF THE DATASETS.

Dataset	#Instances	#Dimension
Letter	*1536*	17
Wbc	*683*	11
Liver disorder	*346*	7
Heberman	*306*	3
Diabetes	*804*	8
Sonar	*208*	60
Heart	*303*	75
Ionosphere	*351*	34
Parkinson	400	22

VI. CONCLUSIONS

In this paper the concept of Support Vector Machines (SVM) and Sliding Mode Control (SMC) technique was first reviewed and then a Sliding Mode Control based Support Vector Machine Radial Basis Function's kernel parameter optimization (SMC-SVM-RBF) method was presented. The proposed method generates significantly higher performance to that of anchor SVM in terms of accuracy and number of support vectors, which implies lower computational complexity. Moreover, the proposed method gives either higher or very competitive performance to the state of the art SVM based techniques such as Zhang's and Zhiliang Liu's methods.

TABLE II. PERFORMANCE OF PROPOSED METHOD VERSUS ANCHOR SVM IN TERM OF ACCURACY AND NUMBER OF SVS.

Dataset	Number of SVs		Accuracy (%) for Test subset		Accuracy (%) for Training subset		fewer SVs
	SVM	Proposed	SVM	Proposed	SVM	Proposed	
Liver disorder	171	158	72.46	73.91	98.79	98.79	7.6
Letter	441	286	92.23	99.68	97.17	100	35.14
Wbc	75	41	99.27	99.27	97.35	97.14	45.33
Heberman	183	119	67.74	72.58	88.12	78.53	34.97
Diabetes	552	471	67.53	66.88	64.13	100	14.67
Sonar	149	149	85.71	85.71	75.83	100	0
Heart	217	217	83.60	91.80	80.18	100	0
Ionosphere	251	135	78.87	95.77	85.71	100	53.78
Parkinson	66	58	92.30	94.87	83.57	97.14	12.12

TABLE III. PERFORMANCE OF PROPOSED METHOD VERSUS ZHANG [16] AND ZHILIANG LIU [12] METHODS.

	Zhang's method		Zhiliang Liu's method		Proposed method	
Dataset	Accuracy (%) for Test subset	Optimal γ value	Accuracy (%) for Test subset	Optimal γ value	Accuracy (%) for Test subset	Optimal γ value
Parkinson	93.51	2.59	93.35	3.61	94.87	0.0006748792225280603
Ionosphere	93.80	5.82	95.23	3.99	95.77	0.059792421140318186
Sonar	83.66	18.31	87.31	5.88	85.71	3.753491664198852
Haberman	71.03	1.59	71.37	1.40	72.58	0.017403710412551898

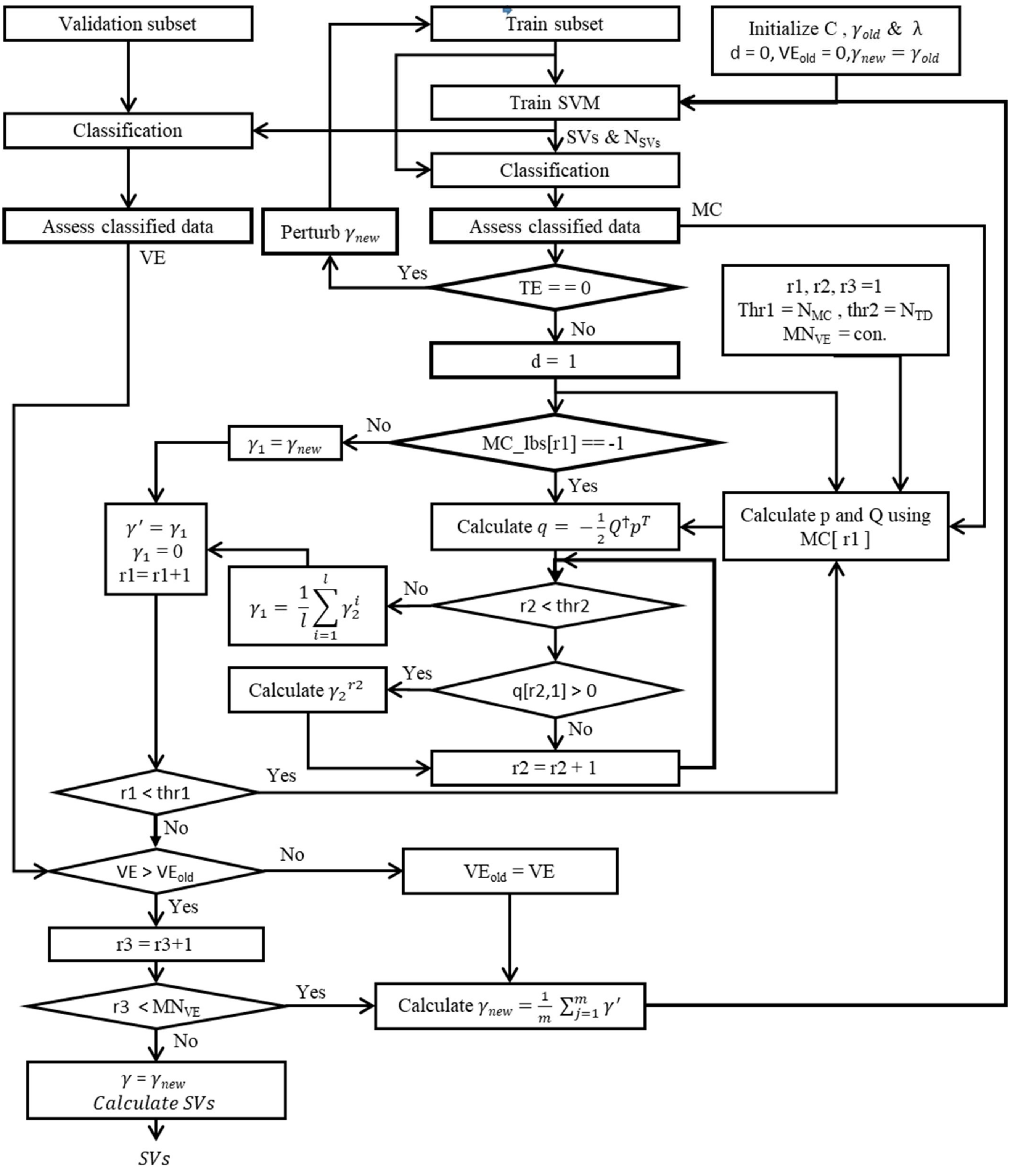

Fig 2. Block diagram of the proposed algorithm.

REFERENCES

[1] C. D. Manning, P. Raghavan, and H. Schütze, “Support vector machines and machine learning on documents,” *Introd. to Inf. Retr.*, pp. 319–348, 2008.

[2] P. Thanh Noi and M. Kappas, “Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using Sentinel-2 imagery,” *Sensors*, vol. 18, no. 1, p. 18, 2018.

[3] M. Famouri, M. Taheri, and Z. Azimifar, “Fast linear svm validation based on early stopping in iterative learning,” *Int. J. Pattern Recognit. Artif. Intell.*, vol. 29, no. 08, p. 1551013, 2015.

[4] L. Auria and R. A. Moro, “Support Vector Machines (SVM) as a Technique for Solvency Analysis,” *SSRN Electronic Journal*, 2008.

[5] S. Demyanov, J. Bailey, K. Ramamohanarao, and C. Leckie, “AIC and BIC based approaches for SVM parameter value estimation with RBF kernels,” in *Asian Conference on Machine Learning*, pp. 97–112, 2012.

[6] A. Abdiansah and R. Wardoyo, “Time complexity analysis of support vector machines (SVM) in LibSVM,” *Int. J. Comput. Appl.*, 2015.

[7] R. Hable and A. Christmann, “On qualitative robustness of support vector machines,” *Journal of Multivariate Analysis*, vol. 102, no. 6, pp. 993–1007, 2011.

[8] T. Joachims, “Text categorization with support vector machines: Learning with many relevant features,” in *European conference on machine learning*, pp. 137–142,1998.

[9] E. Osuna, R. Freund, and F. Girosi, “Training support vector machines: an application to face detection,” in *cvpr*, p. 130,1997.

[10] T. Downs, K. E. Gates, and A. Masters, “Exact simplification of support vector solutions,” *J. Mach. Learn. Res.*, vol. 2, no. Dec, pp. 293–297, 2001.

[11] Y. Li, W. Zhang, and C. Lin, “Simplify support vector machines by iterative learning,” *Neural Inf. Process. Lett. Rev.*, vol. 10, no. 1, pp. 11–17, 2006.

[12] Z. Liu and H. Xu, “Kernel Parameter Selection for Support Vector Machine Classification,” *Journal of Algorithms & Computational Technology*, vol. 8, no. 2, pp. 163–177, 2014.

[13] X.-L. Xia, M. R. Lyu, T.-M. Lok, and G.-B. Huang, “Methods of decreasing the number of support vectors via k-mean clustering,” in *International Conference on Intelligent Computing*, pp. 717–726, 2005.

[14] D. Nguyen and T. Ho, “An efficient method for simplifying support vector machines,” *in Proceedings of the 22nd international conference on Machine learning,* pp. 617–624, 2005.

[15] C. Jose, P. Goyal, P. Aggrwal, and M. Varma, “Local deep kernel learning for efficient non-linear svm prediction,” *in International conference on machine learning,* pp. 486–494, 2013.

[16] D. Zhang, S. Chen, and Z.-H. Zhou, “Learning the kernel parameters in kernel minimum distance classifier,” Pattern Recognition, vol. 39, no. 1, pp. 133–135, 2006.

[17] A Study on Robustness Property of Sliding-Mode Controllers: A Novel Design and Experimental Investigations

[18] Jack Champaigne, Electronics Inc.[Online]. Available: https://www.electronics-inc.com/wp-content/uploads/BenefitsOfClosedLoop.pdf. [Accessed: 10-August-2019].

[19] Z. CHEN, Z. Qian, Z. WANG, X. SHAO, and X. WEN, “Sliding Mode Control Based On SVM for Fractional Order Time-delay System,” DEStech Trans. Eng. Technol. Res., no. icca, 2016.

[20] H.-S. Ahn, Y. Chen, and K. L. Moore, “Iterative Learning Control: Brief Survey and Categorization,” *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1099–1121, 2007.

[21] J. A. K. Suykens, J. Vandewalle, and B. De Moor, “Optimal control by least squares support vector machines,” *Neural networks*, vol. 14, no. 1, pp. 23–35, 2001.

[22] H. Shi, H. Xiao, J. Zhou, N. Li, and H. Zhou, “Radial Basis Function Kernel Parameter Optimization Algorithm in Support Vector Machine Based on Segmented Dichotomy,” *2018 5th International Conference on Systems and Informatics (ICSAI)*, 2018.

[23] D. Geebelen, J. A. K. Suykens, and J. Vandewalle, “Reducing the Number of Support Vectors of SVM Classifiers Using the Smoothed Separable Case Approximation,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 4, pp. 682–688, 2012.

[24] X. Pang, C. Xu, and Y. Xu, “Scaling KNN multi-class twin support vector machine via safe instance reduction,” *Knowledge-Based Systems*, vol. 148, pp. 17–30, 2018.

[25] S. Yin and J. Yin, “Tuning kernel parameters for SVM based on expected square distance ratio,” *Information Sciences*, vol. 370-371, pp. 92–102, 2016.

[26] J.-J. E. Slotine and W. Li, *Applied nonlinear control*, vol. 199, no. 1. Prentice hall Englewood Cliffs, NJ, 1991.

[27] Y. Shtessel, C. Edwards, L. Fridman, and A. Levant, “Introduction: Intuitive theory of sliding mode control,” in *Sliding Mode Control and Observation*, Springer, pp. 1–42. 2014.

[28] B. Gallup “Sliding Mode Control: A Comparison of Sliding Surface Approach Dynamics: http://web.mit.edu/gallup/Public/project.pdf [Mar. 20, 2019].

[29] *Donald Bren School of Information and Computer Sciences @ University of California, Irvine*. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html. [Accessed: 12-May-2019].