Fast Training of Support Vector Machine for Forest Fire Prediction

Steve Chan, Ika Oktavianti Najib*, Verlly Puspita
Center for Research on IOT, Data Science, and Resiliency (CRIDR)
San Diego, CA 92192, USA
*Email: inajib@cridr.org

***Abstract*—Support Vector Machine (SVM) is a binary classification model, which aims to find the optimal separating hyperplane with the maximum margin in order to classify the data. The maximum margin SVM is obtained by solving a convex Quadratic Programming Problem (QPP) and is termed as the hard-margin linear SVM. This optimization problem can solve using commercial Quadratic Programming (QP) code, i.e., Lagrange multipliers. However, the training process is time-consuming. Several decomposition methods have been proposed, which split the problem into a sequence of smaller sub-problems. The Sequential Minimal Optimization (SMO) algorithm is a widely utilized decomposition for SVM. In this paper, SMO algorithm for SVM for regression is utilized to forecast forest fires. Moreover, the Stochastic Gradient Descent (SGD) algorithm is employed for comparison purposes. The comparative results analysis shows that SVR-SMO model outperforms the SGDRegressor model in predicting forest fires.**

Keywords—Fast training of support vector machine; support vector regression; sequential minimal optimization; stochastic gradient descent; forest fire prediction.

I. INTRODUCTION

Support Vector Machine (SVM), as proposed by Vapnik [1] is a supervised learning model that is utilized for classification, regression, and outlier detection problems. SVM implements the structural risk minimization principle, which seeks to minimize the training error and construct confidence intervals for the accuracy. SVM is a robust methodology for solving several classes of problems with small samples, nonlinearity, high dimensionality, and local minimum [2]. SVM has been utilized within several fields, including a feature recognition process, which transforms data from the input space into higher dimensional space, and optimization is performed upon the new vector spaces. This distinguishes SVM from the pattern recognition solution in general, which optimizes the parameters in the transformation results space which is lower than the input space dimension.

SVM has two phases: training and testing, where the training process is the most time-consuming. Training in SVM requires solving a Quadratic Programming (QP) problem. This problem is transformed utilizing the Lagrange multipliers method, and the solution is obtained for finding the set of optimal Lagrange coefficients [3]. Many methods have been proposed to solve the QP problem in the context of faster training.

The majority of SVM training optimization problems are solved utilizing a decomposition algorithm. The proposed decomposition methods lead to faster training, whereby the problem is decomposed more quickly into sub-problems. These decomposition methods repeatedly select a subset of the free variables and optimizes over these variables. One of the utilized decomposition methods is Sequential Minimal Optimization (SMO) proposed by Platt [4]. SMO avoids numerical QP problems and solves the smallest optimization problem at each iteration. Another method for solving optimization problems, which has also been widely utilized for machine learning is that of the Stochastic Gradient Descent (SGD). SGD is an iterative method for optimizing formula use to achieve production goal. In this paper, the SMO algorithm for Support Vector Regression (SVR) is utilized to forecast the problem domain of peatland forest fires. The SGD algorithm is also employed for comparative study with the SVR-SMO model.

Section I provided an introduction. The remainder of this paper is organized as follows. Section II describes related works regarding research implementation of SVM, SMO, etc. Section III provides details regarding the SVM theory, Lagrange Multipliers, Krush-Kuhn-Tucker (KKT) Condition, SVR and Section IV discusses the SVM training method consist of SMO, SGD, and Section V discusses the experiment and results. Finally, interim conclusions are summarized in Section VI.

II. RELATED WORKS

Lin et al. [5] formulated the original SVM problem as the Minimum Enclosing Ball (MEB) approach and proposed SMO for attaining fewer support vectors as well as obtaining an acceptable accuracy compared to the original SVM. The SMO has been modified by the idea of the active set and second order information. The result shows that the proposed method improves the efficiency and reduces memory consumption.

Feng et al. [6] implemented the Modified SMO (MSMO) algorithm of SVM so as to enable and speed up the learning process of the hardware system, via the Integrated Circuit (IC). MSMO is applied with two threshold parameters instead of one. Experimental results show that the designed system has a high detection rate and fast learning process of SVM.

Qihua and Shuai [7] present a new fast SVM learning algorithm for large-scale training set under the condition of sample aliasing. The main idea of this proposed algorithm is that those aliasing training samples, which are not of the same class, are eliminated first, and then the Relative Boundary Vectors (RBVs) are calculated. According to Qihua and Shuai's algorithm, not only the RBVs sample itself, but a near RBVs sample, whose distance to the RBVs is smaller than a certain value, is also selected for SVM training in order to prevent the loss of some critical sample points for the optimal hyperplane. The simulation results demonstrate that this fast learning algorithm is very effective and can be utilized as a pragmatic approach for large-scale SVM training.

Wijnhoven and With [8] evaluated the performance of Stochastic Gradient Descent (SGD) when only a part of the training set is presented to the training algorithm. The SGD algorithm was implemented for learning a linear SVM classifier for object detection. The obtained classification performance of Wijnhoven and With's model is similar to that of state-of-the-art SVM implementations as they are able to obtain a speed up factor in computation time of two or three orders of magnitude.

Cao et al. [9], who solved the problem of fault prediction and failure prognosis for electro-mechanical actuators, utilized SVR. With the large size of sample data, the improved SMO algorithm was employed to solve the SVR model problems. The SMO algorithm is developed by improving stopping criteria as the SVR method can overcome drawbacks of slow convergence and local minimum. The simulation results demonstrate that the SMO-SVR method has characteristics of high prediction accuracy and time efficiency, as well as indicators for preventive measure actions before failure occurs.

Priyadarshini et al. [10] utilized SVR and SMO for link load prediction of a network. SMO was utilized for model training, while SVR was utilized for forecasting. SVR models are robust to parameter variation and can generalize against unseen data and is quite proficient at continuous and adaptive online learning. The result indicates that SVR-SMO performance is quite satisfactory and promising for applications, such as real-time traffic condition prediction.

III. Support Vector Machine Theory

SVM is a binary classification model, which aims to find the optimal separating hyperplane with the maximum margin to separate the involved classes of data (please refer to Figure 1). SVM addresses generalization utilizing a theoretical framework and shows that the generalization error is related to the margin of a hyperplane classifier [11]. This hyperplane is represented by the following equation, where w is called the weight vector, x is the input data, and b is referred to as the bias:

$$H: w^T \cdot x_i + b = 0 \quad (1)$$

$$H_+: w^T \cdot x_i + b = +1 \quad (2)$$

$$H_-: w^T \cdot x_i + b = -1 \quad (3)$$

Since the labels are the same as the {-1, 1} sides of the plane, the constraints can be rewritten as $y(w^T \cdot x + b) \geq 1$ for all training points x with label $y \in \{-1,1\}$(will have one constraint for each training point) [12]. Though the principle of maximum margin is derived through certain inequalities, the larger the margin, the smaller is the probability that a hyperplane will determine the class of a test sample incorrectly [9]. Therefore, the maximum margin of SVM is obtained by solving the following optimization problem:

$$\min_{(w,b)} \frac{1}{2} \|w\|^2 \quad (4)$$

Equation 4 is a convex Quadratic Programming Problem (QPP) and is termed as the hard-margin linear SVM. The formulation may be more succinctly written as:

$$\min_{(w,b)} \frac{1}{2} w^T w \quad (5)$$

$$s.t\ y_i(w^T \cdot x + b) \geq 1,\ \ i = 1, \dots, m$$

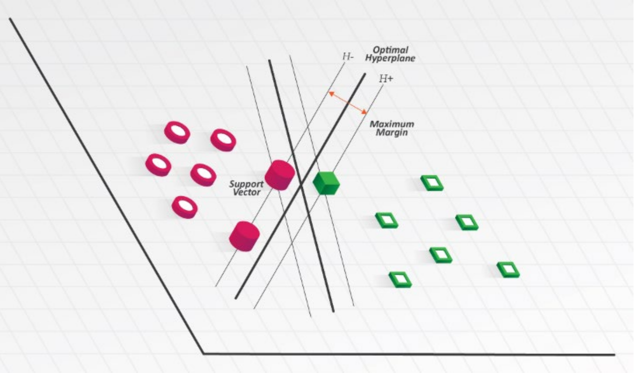

Figure 1. Optimal hyperplane within a two-dimensional space

Figure 1 show the data with two bordering parallel lines that form a margin around central separating line. As a consequence, the algorithm uncovers the elements in the data that touch the margins [13]. These are called the *support vector*. The other elements distanced from the border are not relevant to the solution. Support vectors are found after an optimization step involving a convex quadratic objective and a linear constraint. This optimization problem can then be solved utilizing commercial QP code, i.e., Lagrange multipliers. The method of Lagrange multipliers can handle the inequality constraints and posit the necessary and sufficient conditions for minimizing the primal form of the SVM [14]. With this condition, the primal form turns into an equivalent dual form.

A. Lagrange Multipliers

Lagrange multipliers constitute a mathematical method utilized to solve constrained optimization problems of differentiable functions [15]. One Lagrange multiplier α_i is defined for each constraint, and the constraints $y_i f(x_i) \geq 1$ are re-written as $y_i f(x_i) - 1 \geq 0$. The Langrangian is:

$$\mathcal{L}(w,b,\alpha) = \frac{1}{2}w^T w - \sum_{i=1}^{m} \alpha_i (y_i (w^T \cdot x + b) - 1) \quad (6)$$

Then, $\mathcal{L}$ is differentiated with respect to w, b, and the differential is set to zero:

$$\frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^{m} \alpha_i y_i x_i, \quad (7)$$

$$\frac{\partial L}{\partial b} = 0 \rightarrow b = \sum_{i=1}^{m} \alpha_i y_i = 0 \quad (8)$$

and then the equation of w and b is placed back into the $\mathcal{L}(w,b,\alpha)$ equation in order to eliminate (w,b). Consequently, the Lagrange dual problem is obtained for the original SVM-primal problem.

$$\mathcal{L}(w,b,\alpha) = \max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j {x_i}^T x_j \quad (9)$$

To train the SVM, the feasible region of the dual problem and the maximization of the objective function are necessary and sufficient to specify optimal solution. The optimal solution can then be checked utilizing the Krush-Kuhn-Tucker conditions.

B. *Krush-Kuhn-Tucker Condition*

Although the Lagrange multipliers provide an important optimization technique, it can only be employed under equality constraints, while the SVM minimization problem is restricted by inequalities [16]. In order to tackle the maximal-margin problem, Krush-Kuhn-Tucker (KKT) must be satisfied when performing Lagrange multipliers for inequality constraints. There are five KKT conditions that affect the dual problem [13]:

$$\frac{\partial L}{\partial w} \mathcal{L}(w,b,\alpha) = w - \sum_{i}^{m} \alpha_i y_i x_i = 0 \quad (10)$$

$$\frac{\partial L}{\partial b} \mathcal{L}(w,b,\alpha) = - \sum_{i}^{m} \alpha_i y_i = 0 \quad (11)$$

$$y_i (w^T \cdot x + b) - 1 \geq 0 \quad (12)$$

$$\alpha_i \geq 0 \quad (13)$$

$$\alpha_i (y_i (w^T \cdot x + b)) - 1 = 0 \quad (14)$$

C. *Support Vector Regression (SVR)*

The basic idea of SVR is to find a function $f(x)$ that has at most ϵ from the actualobtained target y_i for all training data [17]. Referring to Figure 2, the region bound by $y_i \pm \epsilon$ is called an ϵ-insensitive tube. The samples deviating from ϵ-insensitive tube can be integrated to the optimization problem by using slack variables (ξ). The error function for SVR can then be written as:

$$\min_{(w,b)} \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{m} (\xi_i^+ + \xi_i^-) \quad (15)$$

Consequently, the dual optimization problem can be rewritten as follows:

$$\max_{\alpha^+,\alpha^-} \left[\sum_{i=1}^{m} (\alpha_i^* - \alpha_i) y_i - \epsilon \sum_{i=1}^{m} (\alpha_i^* - \alpha_i) - \frac{1}{2} \sum_{i,j}^{m} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) \mathbf{x_i} \cdot \mathbf{x_j} \right] \quad (16)$$

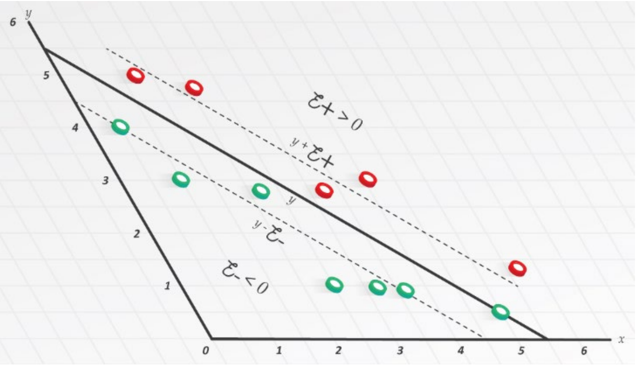

Figure 2. SVR with ϵ-tube

Accordingly, the standard SVR to solve the approximation problem is as follows:

$$f(x) = \sum_{i=1}^{m} (\alpha_i^* - \alpha_i) K(x_i, x) + b \quad (17)$$

where α_i^* and α_i are Lagrange multipliers and $K(x_i, x)$ is a kernel function.

IV. SVM Training method

To reduce the computational complexity of SVM training, the basic and the most commonly used method is to select the most informative samples that have the most possibility to become the support vectors in the training sample set before training the SVM. In this paper, SMO and SGD algorithm are utilized for fast training of SVM. The first review the training procedures of SMO algorithm, and then describe SGD algorithm simply.

A. *Sequential Minimal Optimization (SMO)*

The SMO approach minimizes memory storage for decomposing a large QP problem into a series of smaller QP sub-problems. Each sub-problem is solved analytically to avoid utilizing a time-consuming numerical QP optimization, via optimizing two elements of α_i (Lagrange multipliers).

$$\mathcal{L}(w,b,\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j {x_i}^T x_j \quad (18)$$

$$s.t. 0 \leq \alpha_i \leq C \ \ and \ \sum_{i=1}^{m} \alpha_i y_i = 0$$

where C is a SVM hyper-parameter. Because of the linear equality constraint involving the Lagrange multipliers α_i, the smallest possible problem involves two such multipliers. Then, for any two multipliers $\alpha_1$and α_2, the constraints are reduced to:

$$0 \leq \alpha_1 , \alpha_2 \leq C, \quad (19)$$

$$y_1\alpha_1 + y_2\alpha_2 = k \qquad (20)$$

where k is the negative of the sum over the rest of terms within the equality constraint, which is fixed at each iteration.

B. Stochastic Gradient Descent (SGD)

The SGD algorithm is able to minimize the objective functions that depend upon an integral [18]. SGD has two major steps: individual gradient computation and weight update. SGD continuously fluctuates to converge, where the weight update jumps to a better local minimum for the non-convex error function [19].

Algorithm 1. Stochastic Gradient Descent (SGD)

```
Input: Training data $S$, regularization parameters $\lambda$, learning rate
        $\eta$, initialization $\sigma$
Output: Model parameters $\Theta$= ($w_0$,w, V)
$w_0 \leftarrow 0; w \leftarrow (0, \dots ,0); V \sim N(0,\sigma);$
repeat
    for $(x, y) \in S$ do
        $w_0 \leftarrow w_0 - \eta\left(\frac{\partial}{\partial w_0} l(y(x|\Theta,y) + 2\lambda^0 w_0\right);$
        for $i \in \{1, \dots .p\} \wedge x_i \neq 0$ do
            $w_i \leftarrow w_i - \eta\left(\frac{\partial}{\partial w_i} l(y(x|\Theta,y) + 2\lambda_\pi^w w_i\right);$
            for $f \in \{1, \dots, k\}$ do
                $v_{i,f} \leftarrow v_{i,f} - \eta\left(\frac{\partial}{\partial v_{i,f}} l(y(x|\Theta,y) + 2\lambda_f^v v_{i,f}\right);$
            end
        end
    end
Until stopping criterion is not met;
```

Figure 3. Algorithm Stochastic Gradient Descent

Figure 3 provides a Stochastic Gradient Descent (SGD) algorithm. SGD algorithm tries to find the right weights (w_0 , **w**) by iteratively updating the values of w_0 and **w** by utilizing the value of gradient **V.** The value of the gradient V depends upon the inputs (S), the current values of the model parameter (λ, η, σ) and the cost function **f**. η is the learning rate which determines the size of the steps to reach a minimum, λ is the regularization parameter to reduces overfitting, and σ is standard deviation of sigma. Loss function l ($\hat{y}(x|\Theta, y)$ that measures the cost of prediction $\hat{y}$ when the actual answer is y. The model target is to get the best fit line to predict the value of y based upon the input value x, where x and y is training data sample.

V. Experiment and Result

In order to testify the effectiveness of the algorithms in this paper, the fast training algorithm, SVR-SMO and SGDRegressor, was applied to the peatland forest fire dataset. Further, the performance of SVR-SMO and SGDRegressor (accuracy and the CPU times) will be compared. All reported results are implemented by Python code. Dataset description and experiment results are shown on Section V-A and B respectively.

A. Data Description

The peatland forest fire dataset was obtained from the UCI Machine Learning Repository website [20] and was created by Paulo Cortez and Anibal Morais, University of Minho, Portugal. The meteorological dataset was collected from January 2000 to December 2003. This dataset contains 517 fire observations found in the Montesinho Natural Park in Portugal, 12 attributes of input features, and one output feature representing the total burnt area. This peatland forest fire dataset has multivariate time series features and for regression tasks. The attribute descriptions are given in Table I:

TABLE I. Attribute Description

No	*Attribute*	*Description*
1	X	x-axis coordinate
2	Y	y-axis coordinate
3	Month	Month of the year (a.k.a. month)
4	Day	Day of the week (a.k.a. day)
5	FFMC	Fine Fuel Moisture Code (FFMC) denotes the moisture content surface litter and influences ignition and fire spread
6	DMC	Duff Moisture Code (DMC) represent the moisture content of shallow and deep organic layers, which affect fire intensity
7	DC	Drought Code (DC) for fire intensity
8	ISI	Initial Spread Index (ISI) is a score that correlates with fire velocity spread
9	Temp	Temperature (in Celsius)
10	RH	Relative Humidity (RH) (in %)
11	Wind	Wind Speed (a.k.a. wind) (in km/h)
12	Rain	Rain (in mm/mm^2)
13	area	Total burned area (a.k.a. area) (in ha)

The first four rows denote the spatial and temporal attributes. FFMC, DMC, DC, ISI are the indexes for the Fire Weather Index (FWI) of the Canadian system for rating the fire danger. Temperature, RH, Wind, and Rain constitute meteorological information. Only two geographic features were included, the X and Y axis values, where the fire occurred [21]. Variables of the Total burned area has many 0 values (see the density plot in Figure 4). Consequently, Paulo Cortez and Anibal Morais transformed the variable utilizing the log transformation (log(x+1), where 1 will first be added to the area (to account for the 0 values)).

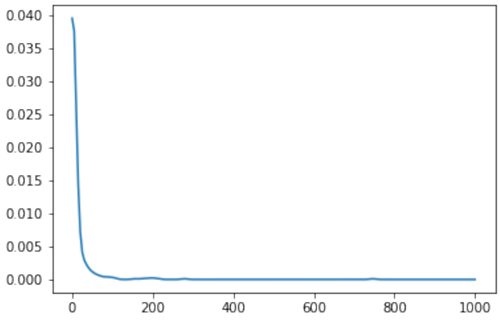

Figure 4. Area burned density

B. *Experimental Results*

SVR-SMO and SGDRegressor were implemented on a Jupyter Notebook utilizing the Python 3 kernel and run atop a machine with Intel(R) Pentium(R) Dual CPU T3400 @2.17 GHz and 2 GB of RAM. 70% and 30% of the employed dataset (517 instances) were selected as the training set and testing set, respectively. Then, both training and test sets were standardized with the StandardScaler function (mean = 0 and standard deviation = 1).

The SVR based upon SMO utilized grid search to optimize the parameters (C and epsilon) and carried out fivefold cross validation for selecting the C value from {0.01, 0.1, 1, 10} and the epsilon value from {10, 1, 0.1, 0.01, 0.001, 0.0001}. As the kernel function, we utilized the Radial Basis Function (RBF) kernel for the SVR kernel defined by $K(x_i, x)= \exp(-||x-y||^2/(2\sigma^2))$. Best parameters obtained by Grid Search are C = 0.01, epsilon = 1, and kernel = RBF.

Parameters of the SGDRegressor include alpha set defaults to 0.0001 to compute the learning rate. The L1 ratio is 0.1, iteration maximum is 1000, epsilon in the epsilon-insensitive loss function is 0.0001, the learning rate schedule is eta 0 = 0.01, the exponent for inverse scaling learning rate is power_t = 0.25, the validation fraction set defaults to 0.1, and the number of iterations with no improvement defaults to 5.

In this study, Root Mean Squared Error (RMSE) are implemented for evaluating prediction performance. RMSE is the square root of the ratio of the quadratic sum of deviations between predicted values and actual values to the times n of prediction. Moreover, the information refers to simulation result comparisons between SVR-SMO and SGDRegressor, as shown in Table 2.

TABLE III. PERFORMANCE COMPARISON OF SVR-SMO AND SGDREGRESSOR

Parameter	Method	
	SVR-SMO	***SGDRegressor***
ε	1	0.001
Max Iteration	1000	1000
CPU Times (s)	0.01639	0.02161
Accurasy (RMSE)	0.66698	3.80567

In Table 2, the results indicate that SVR-SMO exhibits better prediction ability for the UCI forest fire dataset when compared to the SGDRegressor method, while the training speeds for SVR-SMO and SGDRegressor were almost the same.

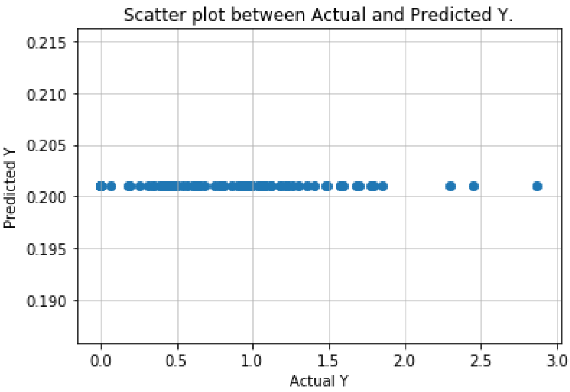

Figure 5. SVR with ϵ-tube

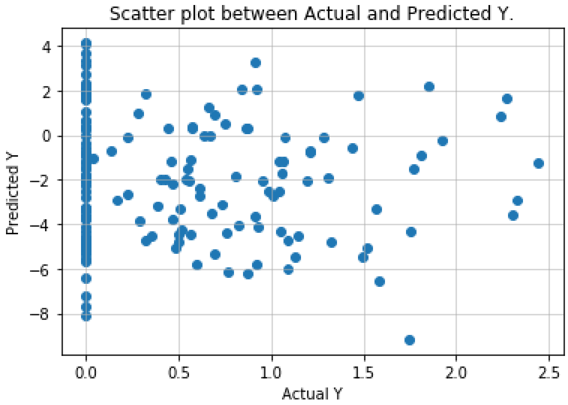

Figure 6. SVR with ϵ-tube

Figure 5 show the scatter plot between actual and predicted value of the burn area part of forest fire dataset utilizing SVR-SMO algorithm. Predicted Y is the estimated outcome or prediction made by the trained model for the given input data and the residual error is describe by e (epsilon). As can be seen in Figure 5, the resulting points form a line that represents the learned relationship between actual and predicted value. In other words, SVR-SMO algorithm reach the goal of regression analysis to fit a line to set of data points. Figure 6 presents scatter plot between actual and predicted value of the burn area part of forest fire dataset utilizing SGDRegressor algorithm. The graph shown that the data points is not linear because the plot of the residual possesses a random distribution. In this case, a line drawn through the data points is horizontal with slop equal to zero.

VI. CONCLUSION AND FUTURE WORK

In this paper, SVR based upon the SMO algorithm is utilized to predict the Total burned area as pertains to a forest fire and compared with the SGDRegressor algorithm. The comparatively small RMSE obtained from the experimental results shows that SVR based upon SMO algorithm has better performance than SGDRegressor, while the training speed for SVR-SMO and SGDRegressor were comparable. Future work will involve studying global convergence of more general decomposition algorithms for multi-objective optimization problems.

ACKNOWLEDGMENT

This research is supported by the Center for Research on IoT, Data Science, and Resiliency (CRIDR), an initiative of Decision Engineering Analysis Laboratory (DEAL) and The International Center for Theoretical Physics (ICTP), a United Nations Educational, Scientific, and Cultural Organization (UNESCO) Category 1 Institution. Sources of data for this research includes the Department of Information Systems, University of Minho, Portugal.

REFERENCES

[1] C. Cortes and V. Vapnik, “Support-Vector Networks,” Machine Learning Journal, vol. 20, Issue 3, pp. 273–297, 1995.

[2] X. Jian-Hua, Z. Xue-gong, and L. Yan-da, “Advance in Support Vector Machine,” *Chinese Control and Decision,* vol. 19, no. 5, pp. 481-484, 2004.

[3] R. A. Hernandez, M. Strum, W. J. Chau, and J. A. Q. Gonzalez, “The Multiple Pairs SMO: A Modified SMO Algorithm for the Acceleration of the SVM Training,” *Proc. of International Joint Conference on Neural Networks*, USA, pp. 1221-1228, 2009.

[4] J. Platt, “Fast training of Support Vector Machines using Sequential Minimal Optimization,” in Advances in Kernel Method-Support Vector Learning, pp. 185-208, 1999.

[5] B. Schdlkopi, C. I. C Burges and A. J. Smala, Editors, MIT Press, Cambridge, MA, pp. 185-208, 1999.

[6] J. Lin, M. Song, and J. Hu, “A SMO approach to Fast SVM for Classification of Large-Scale Data,” *2014 International Conference on IT Convergence and Security (ICITCS),* Beijing, pp. 1-4, 2014.

[7] L. Feng, Z. Li, Y. Wang, C. Zheng, and Y. Guan, “VLSI Design of Modified Sequential Minimal Optimization Algorithm for Fast SVM Training,” *IEEE 2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT),* Hangzhou, pp. 627-629, 2016.

[8] X. Qihua and G. Shuai, “A Fast SVM Classification Learning Algorithm Used to Large Training Set,” *2012 Second International Conference on Intelligent System Design and Engineering Application,* Sanya, Hainan, pp. 15-19, 2012.

[9] R. G. J. Wijnhoven and P. H. N. With, “Fast Training of Object Detection using Stochastic Gradient Descent,” *International Conference on Pattern Recognition,* Istanbul, pp. 424-427, 2010.

[10] Y. Cao, J. Wang, Y. Yu, and R. Xie, “Failure Prognosis for Electro-Mechanical Actuators Based on Improved SMO-SVR Method,” *IEEE Chinese Guidance, Navigation and Control Conference (CGNCC),* Nanjing, pp. 1180-1185, 2016.

[11] D. Priyadarshini, M. Acharya, and A. P. Mishra, “Link Load Prediction Using Support Vector Regression and Optimization,” *International Journal of Computer Applications,* vol. 24, pp. 22-24, 2011.

[12] Jayadeva, R. Kemchandani, and S. Chandra, “Twin Support Vector Machines: Model, Extensions and Applications,” Studies in Computational Intelligence, Springer International Publishing, vol. 659, pp. 1-211, 2017.

[13] B. Wang and V. Pavlu, “Support Vector Machine,” *Based on Notes by Andrew Ng,* 2015.

[14] J. Unpingco, “Python for Probability, Statistic, and Machine Learning,” 2nd edition, Springer International Publishing, pp. 1-384, 2016.

[15] J. Wu, Class Lecture, Topic: “Support Vector Machines,” LAMDA Group, National Key Lab for Novel Software Technology, Nanjing University, China, 2019.

[16] B. T. Smith, Class Lecture, Topic: “Lagrange Multipliers Tutorial in the Context of Support Vector Machine,” Memorial University of Newfoundland, Canada, 2004.

[17] R. F. d. Mello and M. A. Ponti, “Machine Learning: A Practical Approach on the Statistical Learning Theory,” Springer International Publishing, pp. 1-362, 2018.

[18] A. I. Smola and B. Scholkopf, “A Tutorial on Support Vector Regression,” *Statistics and Computing,* vol. 14, pp. 199-222, 2004.

[19] A. G. Carlon, R. H. Lopez, L. F. R. Espath, L. F. F. Miguel, and A. T. Beck, “A Stochastic Gradient Approach for the Reliability Maximization of Passively Controlled Structures,” *Elsevier*, *Engineering Structures,* vol. 186, pp. 1-12, 2019.

[20] A. Sharma, “Guided Stochastic Gradient Descent Algorithm for Inconsistent Datasets,” *Elsevier, Applied Soft Computing Journal,* vol. 73, pp. 1068-1080, 2018.

[21] Paulo Cortez and Anibal Morais, “Forest Fire Dataset,” UCI Machine Learning Repository. February 2008. [Online]. Available from: https://archive.ics.uci.edu/ml/datasets/forest+fires/ 2019.05.23.

[22] Y. Wang, “What Influences Forest Fires Area?” 2016. [Online]. Available from: https://docplayer.net/63027297-What-influences-forest-fires-area-lab-5.html/ 2019.05.23.

ISBN: 978-1-61208-743-6