Large-Scale Fuzzy Least Squares Twin SVMs for Class Imbalance Learning

M. A. Ganaie, M. Tanveer, *Senior Member, IEEE*, and Chin-Teng Lin, *Fellow, IEEE*,
Alzheimer's Disease Neuroimaging Initiative

***Abstract*—Twin support vector machines (TSVMs) have been successfully employed for binary classification problems. With the advent of machine learning algorithms, data have proliferated and there is a need to handle or process large-scale data. TSVMs are not successful in handling large-scale data due to the following: 1) the optimization problem solved in the TSVM needs to calculate large matrix inverses, which makes it an ineffective choice for large-scale problems; 2) the empirical risk minimization principle is employed in the TSVM and, hence, may suffer due to overfitting; and 3) the Wolfe dual of TSVM formulation involves positive-semidefinite matrices, and hence, singularity issues need to be resolved manually. Keeping in view the aforementioned shortcomings, in this article, we propose a novel large-scale fuzzy least squares TSVM for class imbalance learning (LS-FLSTSVM-CIL). We formulate the LS-FLSTSVM-CIL such that the proposed optimization problem ensures that: 1) no matrix inversion is involved in the proposed LS-FLSTSVM-CIL formulation, which makes it an efficient choice for large-scale problems; 2) the structural risk minimization principle is implemented, which avoids the issues of overfitting and results in better performance; and 3) the Wolfe dual formulation of the proposed LS-FLSTSVM-CIL model involves positive-definite matrices. In addition, to resolve the issues of class imbalance, we assign fuzzy weights in the proposed LS-FLSTSVM-CIL to avoid bias in dominating the samples of class imbalance problems. To make it more feasible for large-scale problems, we use an iterative procedure known as the sequential minimization principle to solve the objective function of the proposed LS-FLSTSVM-CIL model. From the experimental results, one can see that the proposed LS-FLSTSVM-CIL demonstrates superior performance in comparison to baseline classifiers. To demonstrate the feasibility of the proposed LS-FLSTSVM-CIL on large-scale classification problems, we evaluate the classification models on the large-scale normally distributed clustered (NDC) dataset. To demonstrate the practical applications of the proposed LS-FLSTSVM-CIL model, we evaluate it for the diagnosis of Alzheimer's disease and breast cancer disease. Evaluation on NDC datasets shows that the proposed LS-FLSTSVM-CIL has feasibility in large-scale problems as it is fast in comparison to the baseline classifiers.**

***Index Terms*—Alzheimer's disease (AD), class imbalance, machine learning, magnetic resonance imaging (MRI), maximum margin, mild cognitive impairment (MCI), pattern classification, structural risk minimization (SRM) principle, support vector machines (SVMs), twin support vector machine (TSVM).**

I. Introduction

The support vector machines (SVMs) [1] are enhanced kernel-based classification algorithms utilizing the concept of margin maximization between the target classes of a classification problem. The SVM has been used across a variety of problems, such as lung nodule classification [2], health care [3], and so on. The SVM ensures that the structural risk minimization (SRM) principle is implemented and, hence, shows better generalization performance. The SVM solves a single quadratic programming problem (QPP); hence, it suffers from large-scale problems. To subsidize the computational complexity, the generalized eigenvalue proximal SVM (GEPSVM) [4] solved the generalized eigenvalue problem in the classification problems. Prompted by the GEPSVM, the twin SVM (TSVM) [5] searches for two proximal planes corresponding to the classes of the binary classification problem by solving two smaller size QPPs. The TSVM is approximately four times faster compared with the SVM. However, the TSVM still requires the calculation of matrix inverses and assumes the existence of nonsingular matrices in the real-world scenarios. However, such an assumption may not be valid in a variety of practical problems. Twin bounded support vector machines (TBSVMs) [6] added an extra regularization term to the primal formulation of the TSVM to

Manuscript received 21 October 2021; revised 17 January 2022 and 1 March 2022; accepted 7 March 2022. Date of publication 23 March 2022; date of current version 1 November 2022. This work was supported in part by the Department of Science and Technology and the Ministry of Electronics and Information Technology, Government of India, through the National Supercomputing Mission under Grant DST/NSM/R&D_HPC_Appl/2021/03.29, in part by the Department of Science and Technology through Interdisciplinary Cyber Physical Systems Scheme under Grant DST/ICPS/CPS-Individual/2018/276, and in part by the Science and Engineering Research Board through the Mathematical Research Impact-Centric Support scheme under Grant MTR/2021/000787. The data used in this study was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI), National Institutes of Health, under Grant U01 AG024904, and Department of Defense ADNI under Grant W81XWH-12-2-0012, whose funding was provided by the National Institute on Aging, National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: F. Hoffmann-La Roche Ltd. and its affiliated company Genentech, Inc.; Janssen Alzheimer Immunotherapy Research and Development, LLC; Merck and Company, Inc.; Alzheimer's Drug Discovery Foundation; Biogen; Bristol-Myers Squibb Company; Meso Scale Diagnostics, LLC; NeuroRx Research; EuroImmun; CereSpir, Inc.; Cogstate; Eisai Inc.; Elan Pharmaceuticals, Inc.; AbbVie, Alzheimer's Association; Fujirebio; GE Healthcare; IXICO Ltd.; Johnson & Johnson Pharmaceutical Research and Development LLC; Eli Lilly and Company; Araclon Biotech; BioClinica, Inc.; Lumosity; Lundbeck; Neurotrack Technologies; Takeda Pharmaceutical Company; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; and Transition Therapeutics. *(Corresponding author: M. Tanveer.)*

M. A. Ganaie and M. Tanveer are with the Department of Mathematics, Indian Institute of Technology Indore, Indore 453552, India (e-mail: phd1901141006@iiti.ac.in; mtanveer@iiti.ac.in).

Chin-Teng Lin is with the Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: Chin-Teng.Lin@uts.edu.au).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TFUZZ.2022.3161729.

Digital Object Identifier 10.1109/TFUZZ.2022.3161729

avoid the issues of overfitting via the SRM principle. Nonparallel SVMs [7] introduced the ϵ-insensitive loss function, and nonparallel support vector regression [8] extended the formulation to ordinal regression. Many modifications of the TSVM have been proposed to address the different kinds of issues in classification problems. Among these issues, the presence of noise and the class imbalance problem are the profound ones.

To subsidize the effect of noise, noisy data points across different applications are assigned smaller fuzzy membership weights via different approaches to reduce their effect, such as in anomaly detection [9], multiview learning [10], and so on. A fuzzy SVM (FSVM) [11] assigned distance-based fuzzy weights to each data point from the class centroid in a manner that the outliers are assigned relatively smaller weights compared to other data samples. For multiclass problems, least squares SVMs [12] defined the fuzzy membership function in a perpendicular direction of the optimal separating hyperplane. A bilateral-weighted FSVM (B-FSVM) [13] and a proximal B-FSVM [14] assigned fuzzy weights by treating the data samples as members of both the classes with different membership weights. An intuitionistic fuzzy TSVM [15] used both the membership and nonmembership weights of a sample for generating the score value to minimize the effect of outliers. An entropy-based fuzzy TSVM [16] generated fuzzy membership values based on the probabilistic neighborhood of each sample, which results in better classification. Asymmetric possibility and necessity regression by the TSVM [17] generate four regression bounds for the regression problems, wherein two functions generate the lower and upper bounds for the possibility model and the other two functions generate the upper and lower bounds of the necessity model for the end regressor. A fuzzy smooth piecewise TSVM [18] uses a polynomial function to solve the unconstrained optimization problem and reduce the effect of outliers via fuzzy membership weights. A kernel-target-alignment-based fuzzy least squares TBSVM [19] uses the kernel target alignment approach for generating the fuzzy values to subsidize the outlier effect. A regularized robust fuzzy TSVM [20] uses the imbalance ratio (IR) in the fuzzy membership. Following this, different approaches, such as within-class-scatter-matrix-based FSVM [21] and Clifford FSVM [22], used Clifford geometric algebra for learning the decision surfaces. For more details, we refer the interested readers to recent review [23].

To subsidize the issues of imbalance class learning wherein the data points of a class are dominated by the data points of another class resulting in the generation of the biased classifier overpowered by the majority data points, fuzzy weights are given to the data samples in binary classification problems. A weighted least squares projection TSVM [24] incorporated the local neighborhood information of the data, and the fuzzy least squares TSVM used the least squares concept to deal with the class imbalance problems [25]. A multilevel weighted SVM [26] uses a cost-sensitive multilevel framework and an expected maximization imputation method in imbalanced health care data. A fuzzy total-margin-based SVM [27] incorporated the total margin concept with different cost functions to deal with the class imbalance problem. Other models handling the imbalance problem include weighted least squares TSVM [28], twin sphere SVMs [29], and fuzzy TSVM [30]. For the datasets with different IRs, the aforementioned models for class imbalance use the fuzzy membership function, whose range is fixed irrespective of change in the IR. To give appropriate weights to the majority class, a robust fuzzy least squares TSVM for class imbalance learning (RFLSTSVM-CIL) [31] used the fuzzy membership functions, whose range is a function of the IR. The RFLSTSVM-CIL incorporated the IRs of the data samples into the fuzzy membership function to assign the appropriate weights to the data samples of the imbalance datasets.

Analyzing the existing TSVM [5] and RFLSTSVM-CIL [31], there are several shortcomings, which are the following:

1) TSVM-based models involve matrix inverse computation, which may be intractable in large-scale problems, and also, singularity issues may arise.
2) The TSVM and the RFLSTSVM-CIL are based on the empirical risk minimization principle and assume that the matrices appearing in the dual formulation are always well conditioned. However, in real-world problems, this assumption may not be satisfied. The empirical risk minimization may lead to overfitting issues and result in lower generalization.
3) The RFLSTSVM-CIL and the TSVM solve the linear and nonlinear cases separately as the nonlinear case is different from the linear case when linear kernel is used.

To overcome the aforementioned issues, we formulated a novel large-scale fuzzy least squares TSVM for class imbalance learning (LS-FLSTSVM-CIL). The advantages of the LS-FLSTSVM-CIL are as follows:

1) The LS-FLSTSVM-CIL reformulates the primal form of the RFLSTSVM-CIL to make it feasible for large-scale datasets.
2) The LS-FLSTSVM-CIL minimizes the structural risk, which is the marrow of statistical learning. The SRM principle overcomes the issues of overfitting and, hence, results in better generalization performance.
3) The LS-FLSTSVM-CIL avoids the computation of matrix inverse, whereas other given baseline models need to calculate the matrix inverse, which is infeasible for large-scale problems.
4) For the nonlinear case, the given baseline models employ kernel-generated surfaces, whereas the proposed LS-FLSTSVM-CIL directly incorporates the kernel trick into the dual formulation.
5) The LS-FLSTSVM-CIL handles the class imbalance problems unlike the TSVM and the LSTSVM via fuzzy membership weights. The fuzzy membership gives appropriate weights to the samples of majority class based on the sample distance and the IR of the data.
6) Evaluation on benchmark datasets demonstrates the applicability of the proposed LS-FLSTSVM-CIL on a wide range of datasets from small-scale to large-scale data.
7) To demonstrate the efficacy of the proposed LS-FLSTSVM-CIL model in practical applications, we use the model for the diagnosis of Alzheimer's disease (AD) and breast cancer disease.

The rest of this article is organized as follows. Section II gives the background of the previous work. Section IV discuss the SMO algorithm. Sections III and V give the details of the proposed model and its computational complexity, respectively. Experimental results are given in Section VI. The applications of the proposed model are presented in Section VII. Finally, Section IX concludes this article.

II. Related Work

Consider $T = \{(x_i, y_i) \mid x_i \in \mathbb{R}^n,\ y_i \in \{-1, +1\}\}\ \forall i = 1, 2, \ldots, l$. Here, l represents the number of samples in a dataset T. Each vector is treated as the column vector unless transposed (t) explicitly. $\|.\|$ denotes the 2-norm, and IR is the ratio of the number of majority class samples to the number of minority class samples. Here, we assume that the positive class represents the minority class, and the negative class represents the majority class. Mathematically

$$\text{IR} = \frac{\text{Count of data points in the majority class}}{\text{Count of data points in the minority class}}. \tag{1}$$

We will evaluate the proposed classification model with TSVM [5], LSTSVM [32], fuzzy TSVM [5], [33], and RFLSTSVM-CIL [31]. In this section, we discuss the RFLSTSVM-CIL [31] model. Let $A \in \mathbb{R}^{m_1 \times n}$ denote the positive data points and $B \in \mathbb{R}^{m_2 \times n}$ denote the negative data points of a binary class problem.

A. Robust Fuzzy Least Squares TSVM for Class Imbalance Learning

The optimization problem of the nonlinear RFLSTSVM-CIL is

$$\begin{aligned}\min_{w_1, b_1, \xi_2} \quad & \frac{1}{2}\|K(A, C^t)\, w_1 + eb_1\|^2 + \frac{c_1}{2}(S_2\xi_2)^t(S_2\xi_2) \\ \text{s.t.} \quad & -(K(B, C^t)\, w_1 + eb_1) + \xi_2 = e \end{aligned} \tag{2}$$

and

$$\begin{aligned}\min_{w_2, b_2, \xi_1} \quad & \frac{1}{2}\|K(B, C^t)\, w_2 + eb_2\|^2 + \frac{c_2}{2}(S_1\xi_1)^t(S_1\xi_1) \\ \text{s.t.} \quad & (K(A, C^t)\, w_2 + eb_2) + \xi_1 = e \end{aligned} \tag{3}$$

where $C = [A; B]$, slack variables are denoted by ξ_i, S_i are the fuzzy membership weights, e is the appropriate dimension vector of ones, $K(.)$ is the kernel function, c_i are the penalty parameters, and $[w_i, b_i]$ are the hyperplane parameters, for $i = 1, 2$.

The first term in (2) and (3) sums the squared distances from the hyperplanes to the points of the corresponding class. Thus, the minimization of first terms makes the hyperplanes proximal to the corresponding class. The constraints ensure that the hyperplane is at a distance of 1 from the fuzzy weighted points of another class. The vector of error variables gives the error when the hyperplane is closer than this minimum distance of 1. Hence, the second term minimizes the sum of error variables and, hence, attempts to minimize the misclassification due to the samples of another class.

III. Proposed Large-Scale Fuzzy Least Squares TSVM for Class Imbalance Learning

To reduce the imbalance problem wherein the optimal hyperplane is effected by the majority class, the data samples are assigned weights based on the fuzzy membership function [31]. The positive samples are assigned 1 as the fuzzy membership, whereas the fuzzy membership of negative data points is given by

$$\begin{aligned}\text{mem} = {} & \frac{1}{1 + \text{IR}} \\ & + \left(\frac{\text{IR}}{1 + \text{IR}}\right)\left(\frac{\exp(c_0(\frac{D_1 - D_2}{D} - \frac{D_2}{R_2})) - J}{\exp(c_0) - J}\right)\end{aligned} \tag{4}$$

where IR is the imbalance ratio [as given in (1)], D_1 and D_2 are Euclidean distances from the centroids of the positive class and the negative class, respectively, D is an Euclidean distance between the two centroids, R_2 is the maximum distance among the samples of the negative class from its centroid, c_0 is the scale of the exponential function, and $J = \exp(-2c_0)$.

The optimization problems of the proposed method for linear and nonlinear cases are given as follows:

A. Linear Case

The proposed LS-FLSTSVM-CIL seeks a pair of hyperplanes (f_1 and f_2) as follows:

$$f_1(x) = w_1^t x + b_1 = 0 \ \text{ and } \ f_2(x) = w_2^t x + b_2 = 0 \tag{5}$$

where $w_i \in \mathbb{R}^n, x \in \mathbb{R}^n$, and $b_i \in \mathbb{R}$, for $i = 1, 2$.

The optimization problem of the proposed LS-FLSTSVM-CIL is

$$\begin{aligned}\min_{w_1, b_1, \eta_1, \xi_2} \quad & \frac{c_3}{2}(\|w_1\|^2 + b_1^2) + \frac{1}{2}\eta_1^t\eta_1 + \frac{c_1}{2}(S_2\xi_2)^t(S_2\xi_2) \\ \text{s.t.} \quad & Aw_1 + eb_1 = \eta_1 \\ & -(Bw_1 + eb_1) + \xi_2 = e \end{aligned} \tag{6}$$

and

$$\begin{aligned}\min_{w_2, b_2, \eta_2, \xi_1} \quad & \frac{c_4}{2}(\|w_2\|^2 + b_2^2) + \frac{1}{2}\eta_2^t\eta_2 + \frac{c_2}{2}(S_1\xi_1)^t(S_1\xi_1) \\ \text{s.t.} \quad & Bw_2 + eb_2 = \eta_2 \\ & (Aw_2 + eb_2) + \xi_1 = e \end{aligned} \tag{7}$$

where $[w_i \ \ b_i]$ are the hyperplane parameters, ξ_i are the slack variables, S_i denote the fuzzy membership weights, e denotes the vector of ones with appropriate dimensions, and c_j are the positive hyperparameters, for $i = 1, 2$ and $j = 1, 2, 3, 4$.

The LS-FLSTSVM-CIL model finds two hyperplanes in a manner that each hyperplane is closer to the data points of one class and as far as possible from the data points of another class. The first term in the objective functions (6) and (7) implements the SRM principle. In the SVM, the SRM principle is implemented by margin maximization between the two classes, which is measured in terms of the Euclidean distance between the two supporting hyperplanes. The corresponding margin for (6) is given by the distance between the proximal planes $w_1^t x + b_1 = 0$

and $w_1^t x + b_1 = -1$. The one-side margin between these two planes with respect to $w_1^t x + b_1 = 0$ is given by $\frac{1}{\sqrt{\|w_1\|^2+b_1^2}}$. Transform $x \in \mathbb{R}^n$ to $\mathbb{R}^{n+1}$ space as $\boldsymbol{x} = [x^t, 1]^t$. Hence, the proximal and bounding hyperplanes are given by $\boldsymbol{w_1^t x} = [w_1, b_1]^t[x^t, 1] = 0$ and $\boldsymbol{w_1^t x} = [w_1, b_1]^t[x^t, 1] = -1$, respectively. The margin between these two planes is $\frac{1}{\sqrt{\|w_1\|^2+b_1^2}}$; hence, the SRM principle is implemented. The second term minimizes the sum of squared distances of each class from the corresponding hyperplanes such that each plane is closer to its class, and the last term in the objective function penalizes the samples as per their distance from the class centroids via fuzzy membership weights, and the minimization of this term leads to the hyperplane to be at a distance of 1 from the points of other class. Each sample is weighted by the fuzzy membership, which results in a model that is more robust and, hence, has better generalization performance.

Unlike the RFLSTSVM-CIL [31], the proposed LS-FLSTSVM-CIL introduces an extra regularization terms $\frac{c_3}{2}(\|w_1\|^2 + b_1^2)$ and $\frac{c_4}{2}(\|w_2\|^2 + b_2^2)$ [in QPPs (6) and (7)] in the RFLSTSVM-CIL, which embodies the principle of SRM and, hence, avoids the issues of overfitting. We also added one constraint in each of the optimization problems [first constraint in (6) and (7)]. The modified Lagrangian of the proposed LS-FLSTSVM-CIL formulation is such that the large matrix inversion calculations are no longer needed, which render the proposed LS-FLSTSVM-CIL model feasible for large-scale problems.

The Lagrangian of the optimization problem (6) is

$$\begin{aligned} L = &\frac{c_3}{2}(\|w_1\|^2 + b_1^2) + \frac{1}{2}\eta_1^t\eta_1 + \frac{c_1}{2}(S_2\xi_2)^t(S_2\xi_2) \\ &+ \alpha^t(\eta_1 - Aw_1 - eb_1) \\ &+ \beta^t(\xi_2 - Bw_1 - eb_1 - e) \end{aligned} \tag{8}$$

where α and β are the Lagrangian multipliers.

Applying the Karush–Kuhn–Tucker (KKT) conditions, we have

$$c_3 w_1 - A^t\alpha - B^t\beta = 0 \tag{9}$$

$$c_3 b_1 - e^t\alpha - e^t\beta = 0 \tag{10}$$

$$\eta_1 + \alpha = 0 \tag{11}$$

$$c_1 S_2^t(S_2\xi_2) + \beta = 0 \tag{12}$$

$$Aw_1 + eb_1 = \eta_1 \tag{13}$$

$$-(Bw_1 + eb_1) + \xi_2 = e. \tag{14}$$

Rewriting (9) and (10) in matrix form, we have

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = \frac{1}{c_3}\begin{bmatrix} A^t & B^t \\ e^t & e^t \end{bmatrix}\begin{bmatrix} \alpha \\ \beta \end{bmatrix}. \tag{15}$$

From (15), one can see that the hyperplane $[w_1; b_1]$ is obtained without the matrix inversion. Hence, the proposed LS-FLSTSVM-CIL avoids the matrix invertibility leading to its scalability to large-scale problems.

Substituting (11), (12), and (15) into (8), we obtain the unconstrained dual problem as

$$\max_{\alpha,\beta} -\frac{1}{2}(\alpha^t \;\; \beta^t)\,\hat{Q}\,(\alpha^t \;\; \beta^t)^t - c_3\beta^t e \tag{16}$$

where

$$\hat{Q} = \begin{bmatrix} AA^t + c_3 I & AB^t \\ BA^t & BB^t + \frac{c_3}{c_1}(S_2^{-1})^2 \end{bmatrix} + E$$

and the Wolfe dual of the optimization problem (7) is

$$\max_{\lambda,\theta} -\frac{1}{2}(\lambda^t \;\; \theta^t)\,Q'\,(\lambda^t \;\; \theta^t)^t - c_4\theta^t e \tag{17}$$

where

$$Q' = \begin{bmatrix} BB^t + c_4 I & BA^t \\ AB^t & AA^t + \frac{c_4}{c_2}(S_1^{-1})^2 \end{bmatrix} + E$$

where S_i is the diagonal matrix with the fuzzy membership weights along the diagonal (for $i = 1, 2$), E is a matrix of ones, and I is the appropriate dimension identity matrix. From the optimization problems (16) and (17), it is evident that no matrix inversions are involved, which enables the scalability of the proposed LS-FLSTSVM-CIL model to large-scale problems.

Testing data point x is assigned the class as $\text{class}(x) = \underset{i=1,2}{\text{argmin}}\ (|x^t w_i + b_i|)$, where $|.|$ denotes the perpendicular distance from the optimal hyperplane.

B. Nonlinear Case

To deal with the nonlinear cases, the existing models, i.e., TSVM, LSTSVM, FTSVM, and RFLSTSVM-CIL, use kernel-generated surfaces; however, the proposed LS-FLSTSVM-CIL introduces the direct kernel function $K(x, y) = \phi(x)^t\phi(y)$ in the linear case, which enacts the Hilbert space transformation $\mathbf{x} = \phi(x)$, where $\mathbf{x} \in \mathcal{H}, \mathcal{H}$ is the Hilbert space. Hence, the training dataset is given by $\mathbf{T} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l)\}$.

The optimization problem of the proposed nonlinear LS-FLSTSVM-CIL in the Hilbert space $\mathcal{H}$ is

$$\begin{aligned} \min_{w_1, b_1, \eta_1, \xi_2} \;& \frac{c_3}{2}(\|w_1\|^2 + b_1^2) + \frac{1}{2}\eta_1^t\eta_1 + \frac{c_1}{2}(S_2\xi_2)^t(S_2\xi_2) \\ \text{s.t. } & \phi(A)w_1 + eb_1 = \eta_1 \\ & -(\phi(B)w_1 + eb_1) + \xi_2 = e \end{aligned} \tag{18}$$

and

$$\begin{aligned} \min_{w_2, b_2, \eta_2, \xi_1} \;& \frac{c_4}{2}(\|w_2\|^2 + b_2^2) + \frac{1}{2}\eta_2^t\eta_2 + \frac{c_2}{2}(S_1\xi_1)^t(S_1\xi_1) \\ \text{s.t. } & \phi(B)w_2 + eb_2 = \eta_2 \\ & (\phi(A)w_2 + eb_2) + \xi_1 = e. \end{aligned} \tag{19}$$

Here, the terms in the above optimization problems have the same goal as given in the linear case.

The Lagrangian of the optimization problem (18) is

$$L = \frac{c_3}{2}(\|w_1\|^2 + b_1^2) + \frac{1}{2}\eta_1^t\eta_1 + \frac{c_1}{2}(S_2\xi_2)^t(S_2\xi_2)$$

$$
\begin{aligned}
&+\alpha^t(\eta_1-\phi(A)w_1-eb_1)\\
&+\beta^t(\xi_2-\phi(B)w_1-eb_1-e)
\end{aligned}
\tag{20}
$$

where α and β are the Lagrangian multiplier vectors.

Applying the KKT conditions, we have

$$c_3w_1-\phi(A)^t\alpha-\phi(B)^t\beta=0 \tag{21}$$

$$c_3b_1-e^t\alpha-e^t\beta=0 \tag{22}$$

$$\eta_1+\alpha=0 \tag{23}$$

$$c_1S_2^t(S_2\xi_2)+\beta=0 \tag{24}$$

$$\phi(A)w_1+eb_1=\eta_1 \tag{25}$$

$$-(\phi(B)w_1+eb_1)+\xi_2=e. \tag{26}$$

Rewriting (21) and (22) in matrix notation, we have

$$\begin{bmatrix} w_1\\ b_1\end{bmatrix}=\frac{1}{c_3}\begin{bmatrix}\phi(A)^t & \phi(B)^t\\ e^t & e^t\end{bmatrix}\begin{bmatrix}\alpha\\ \beta\end{bmatrix}. \tag{27}$$

From (27), one can see that the hyperplane $[w_1;b_1]$ is obtained without the matrix inversion. Hence, the proposed LS-FLSTSVM-CIL avoids the matrix invertibility leading to its scalability to large-scale problems.

Substituting (23), (24), and (27) into (20), we have the unconstrained dual problem as

$$\max_{\alpha,\beta}-\frac{1}{2}(\alpha^t\ \ \beta^t)\,\hat{Q}\,(\alpha^t\ \ \beta^t)^t-c_3\beta^te \tag{28}$$

where

$$\hat{Q}=\begin{bmatrix}K(A,A^t)+c_3I & K(A,B^t)\\ K(B,A^t) & K(B,B^t)+\frac{c_3}{c_1}(S_2^{-1})^2\end{bmatrix}+E$$

and the Wolfe dual of the optimization problem (19) is

$$\max_{\lambda,\theta}-\frac{1}{2}(\lambda^t\ \ \theta^t)Q'(\lambda^t\ \ \theta^t)^t-c_4\theta^te \tag{29}$$

where

$$Q'=\begin{bmatrix}K(B,B^t)+c_4I & K(B,A^t)\\ K(A,B^t) & K(A,A^t)+\frac{c_4}{c_2}(S_1^{-1})^2\end{bmatrix}+E$$

where S_i is the diagonal matrix with the fuzzy membership weights along the diagonal (for $i=1,2$), E is a matrix of ones, and I is the appropriate dimension identity matrix.

The pair of nonparallel optimal hyperplanes obtained after solving the optimization problems (28) and (29) is

$$f_1(x)=\frac{1}{c_3}\Big[K(x,A^t),\ K(x,B^t)\Big]\begin{bmatrix}\alpha\\ \beta\end{bmatrix}+b_1 \tag{30}$$

where $b_1=\frac{1}{c_3}(e^t\alpha+e^t\beta)$ and

$$f_2(x)=\frac{1}{c_4}\Big[K(x,B^t),\ K(x,A^t)\Big]\begin{bmatrix}\lambda\\ \theta\end{bmatrix}+b_2 \tag{31}$$

where $b_2=-\frac{1}{c_4}(e^t\lambda+e^t\theta)$.

The test data point x is assigned the label as $\text{class}(x)=\underset{i=1,2}{\text{argmin}}\ (|f_i(x)|)$, where $|.|$ denotes the perpendicular distance from the optimal hyperplane.

IV. SMO for Large-Scale Fuzzy Least Squares TSVM for Class Imbalance Learning

Solving the linear system of equations via the matrix inversion process limits the applicability of least squares sense problems to large-scale problems. For large-scale data, we follow the iterative procedure to optimize the Wolfe dual of the proposed formulation. Several approaches have been followed in the literature, such as CVX solver [34], [35]; however, we follow a sequential minimal optimization (SMO) [36], [37] approach to solve the proposed optimization problems.

We modify the optimization problems (28) and (29) as follows:

$$\max_{\alpha,\beta}-\frac{1}{2}(\alpha^t\ \ \beta^t)\hat{Q}(\alpha^t\ \ \beta^t)^t-c_3(\mathbf{0}^t\ \ e^t)(\alpha^t\ \ \beta^t)^t \tag{32}$$

where

$$\hat{Q}=\begin{bmatrix}K(A,A^t)+c_3I & K(A,B^t)\\ K(B,A^t) & K(B,B^t)+\frac{c_3}{c_1}(S_2^{-1})^2\end{bmatrix}+E$$

and

$$\max_{\lambda,\theta}-\frac{1}{2}(\lambda^t\ \ \theta^t)Q'(\lambda^t\ \ \theta^t)^t-c_4(\mathbf{0}^t\ \ e^t)(\lambda^t\ \ \theta^t)^t \tag{33}$$

where

$$Q'=\begin{bmatrix}K(B,B^t)+c_4I & K(B,A^t)\\ K(A,B^t) & K(A,A^t)+\frac{c_4}{c_2}(S_1^{-1})^2\end{bmatrix}+E$$

where $\mathbf{0}$ and e represent the appropriate dimensional column vectors of zeros and ones, respectively.

Rewriting the general formulation from (32) and (33), we have

$$f(z)=\max_{z}-\frac{1}{2}z^t\tilde{Q}z-v^tz \tag{34}$$

where z is a vector of length l, $\tilde{Q}$ is the positive-definite square matrix of size $l\times l$, and $v\geq 0$ is a vector of appropriate dimensions.

The iterative procedure used to optimize the unconstrained optimization problem (34) is given in Algorithm 1. Here, for the derivative of the optimization problem (34) and for every updated vector z, we check the optimal condition (the detailed calculations of the derivative of the optimization problem is given in the supplementary material). The vector z is said to be optimal if the optimal condition is less than the tolerance value.

Theorem 1: The sequence of z^n obtained via SMO converges to the global optima of (34).

Proof: Following the definition of $\|F(z^n)\|^2$, where $F(z^n)=(F_1(z^n),F_2(z^n),\ldots,F_l(z^n))$, we have

$$\|F(z^{n+1})\|^2-\|F(z^n)\|^2=-p^2\sum_{j=1}^{l}\tilde{Q}(j,j) \tag{35}$$

Algorithm 1: Sequential Minimal Optimization.

```
Result: Optimal $z$
Set $n = 0$ and initialize the $z$ as $z^n = z^0$, where
  $z^n = (z_1^n, z_2^n, \cdots, z_l^n)$;
Compute $F_i(z^0) = \frac{\partial f}{\partial z_i^0}$;
while $z$ is not optimal do
    Compute $i = argmax_j \ \frac{F_j^2}{2\tilde{Q}(j,j)}$;
    Update $z_i^n$ as $z_i^{n+1}$ and $z^n \rightarrow z^{n+1}$ ;
    Compute $F_i(z^{n+1}) = \frac{\partial f}{\partial z_i^{n+1}}, \forall i$;
end
```

where p is the step size. Equation (35) is always negative due to positive-definite kernel $\tilde{Q}(j, j) \geq 0 \, \forall j$.

Substituting $p^2 = \|F(z^{n+1})\|^2 - \|F(z^n)\|^2$ into (35), we have

$$\|F(z^{n+1})\|^2 - \|F(z^n)\|^2 = -\|z^{n+1} - z^n\|^2 \sum_{j=1}^{l} \tilde{Q}(j, j). \tag{36}$$

Equality (36) results in $\{\|F(z^n)\|^2\}$, which is a decreasing sequence with lower bounded by zero in the Euclidean space as $\|F(z^n)\|^2 \geq 0$. Hence, $\|F(z^n)\|^2$ is bounded below by zero and is monotonically decreasing. From (36), $\{z^n\}$ results in a Cauchy sequence.

In addition, $\|F(z^n)\|^2$ is the positive-definite quadratic form as $F_j^2 \ \forall j$ is the positive definite. Thus, the set $Z = \{z | \|F(z)\|^2 \leq \|F(z^0)\|^2\}$ is a compact set. The infinite bounded sequence, $\{z^n\} \subseteq Z$, has a convergent subsequence in Z. Let, for some $k \in \mathbb{N}$, $\{z^{n_k}\}$ be the convergent subsequence of $\{z^n\}$ with the limit point $\tilde{z}$. Now, $\lim_{k\to\infty} z^{n_k} = \tilde{z}$ implies that $\lim_{k\to\infty} F_j^2(z^{n_k}) = F_j^2(\tilde{z}) \, \forall j$. Hence, $\lim_{k\to\infty} \|F(z^{n_k})\|^2 = 0$ as $\{\|F(z^n)\|^2\}$ is a decreasing sequence. We know that $0 \leq F_j(z^{n_k}) \leq F(z^{n_k}) \, \forall j$; therefore, $\lim_{k\to\infty} F_j^2(z^{n_k}) = F_j^2(\tilde{z}) = 0$. Hence, $F_j(\tilde{z}) = 0 \ \forall j$. With KKT conditions, the optimal solution of (34) is $\tilde{z}$. Hence, (34) has a unique global optima z^* as $F(z)$ is a strictly convex function.

To prove the subsequence $\{z^{n_k}\}$, $k \in \mathbb{N}$ converges to z^*. Suppose that the sequence does not converge to z^*. Then, there exists some ϵ neighborhood such that $\|z^{n_k} - z^*\|^2 \geq \epsilon$ for all but finitely many terms. z^* is the unique global optima of (34); hence, there is contradiction. $\tilde{z}$ is shown to be the global optima of (34). This completes the proof. ■

V. Computational Complexity

Here, we evaluate the computational complexity of the baseline classifiers and the proposed LS-FLSTSVM-CIL classifier. Suppose that m_1 is the number of positive data points and m_2 is the number of negative data points with $l = m_1 + m_2$. Here, we discuss the complexity of the models for optimization problems in the nonlinear case. The Sherman–Morrison–Woodbury (SMW) formulation is used while calculating the inverses in the LSTSVM and the RFLSTSVM-CIL. Using SMW, three inverses of smaller size are calculated. Hence, the time complexity of the LSTSVM and FLSTSVM-CIL models involves two inversions of order $O(m_1^3)$ and one inversion of size $O(m_2^3)$ if $m_1 < m_2$, otherwise two inversions of order $O(m_2^3)$ and one inversion of size $O(m_1^3)$. Both the TSVM and the FTSVM involve the optimization of the QPP. Hence, the computational complexity of each optimization problem in the TSVM and the FTSVM is approximately $O((\frac{l}{2})^3)$. The proposed formulation involves the generation of fuzzy membership weights and the SMO algorithm for the optimization of the unconstrained optimization problem. In traditional fuzzy membership functions, which assign weights to all the samples, the complexity is $O(l)$. However, the proposed formulation assigns fuzzy weights to negative class samples (majority class). Hence, the complexity of weight generation is $O(m_2)$ with $m_2 < l$. The SMO algorithm breaks the large QPP into small QPPs, which are solved analytically. The complexity of the SMO algorithm is $O(l)$ to $O(l^{2.2})$. Thus, the complexity of each optimization problem in the proposed LS-FLSTSVM-CIL model lies approximately between $O(l + m_2) \approx O(l)$ and $O(l^{2.2} + m_2) \approx O(l^{2.2})$.

VI. Experimental Results

Here, the performance evaluation of the baseline classifiers and the proposed LS-FLSTSVM-CIL classifier is done. The data are taken from UCI [38], KEEL [39], and normally distributed clustered (NDC) [40] repositories. For NDC datasets, the feature size of each data point is 10. Experimental analysis is performed using MATLAB R2017a on the PC with the system specifications of 2X Intel Xeon processor (2.3 GHz) with 128-GB RAM. Gaussian kernel $K(x_1, x_2) = e^{-\mu\|x_1 - x_2\|^2}$ is followed for the nonlinear case. The proposed LS-FLSTSVM-CIL model is solved using the SMO algorithm. The tolerance parameter of the SMO algorithm for UCI [38], KEEL [39], and NDC [40] datasets is fixed with $\epsilon = 0.0001$. For the faster convergence of the SMO algorithm, we empirically set the number of iterations to 10^2 for UCI and KEEL datasets and 10^4 for NDC datasets. The data are randomly divided into training and testing sets with 70:30 ratio, where 70% samples are evaluated for training and 30% samples are evaluated for testing. The optimal parameters corresponding to different models are chosen using fivefold cross validation from the following ranges: $c_0 = [0.5, 1, 1.5, 2, 2.5]$, $\mu = c_i = [10^{-5}, 10^{-4}, \ldots, 10^4, 10^5]$ (for $i = 1, 2, 3, 4$).

The following parameters are used to evaluate the performance:

$$\text{Accuracy, AUC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{37}$$

$$\text{Sensitivity or Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{38}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{39}$$

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{40}$$

$$\text{G-mean} = \sqrt{\text{Precision} \times \text{Recall}} \tag{41}$$

where false positive, true positive, false negative, and true negative are denoted by FP, TP, FN, and TN, respectively.

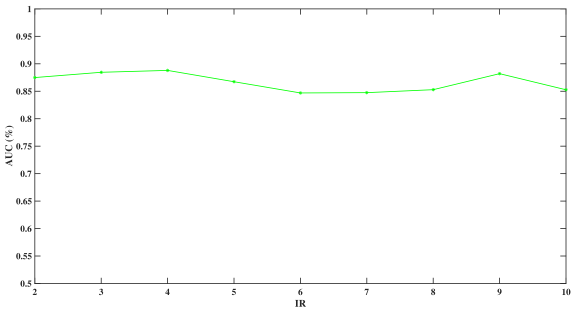


Fig. 1. Performance of the proposed LS-FLSTSVM-CIL model with the varying IR on the NDC dataset of size $90k$.

A. Experiments on UCI [38] and KEEL [39] Datasets

Table I gives the detailed performance of the classifiers on 32 UCI and KEEL benchmark datasets. It is evident that the proposed LS-FLSTSVM-CIL classifier demonstrated superior average performance in rank and accuracy. In the aus dataset, the proposed LS-FLSTSVM-CIL classifier shows approximately 87.53% accuracy in contrast to other classifiers, which show less than 85.5% accuracy. In the ecoli-0-4-6_vs_5 dataset, the proposed LS-FLSTSVM-CIL classifier showed approximately 6% more accuracy in contrast to TSVM, FTSVM, and RFLSTSVM-CIL classifiers. Similarly, in checkerboard_Data, ecoli0137vs26, iono, and segment0, the proposed LS-FLSTSVM-CIL model achieved more accuracy in contrast to other baseline classifiers. Moreover, we performed the overall win–tie–loss analysis on all the classification models based on 32 datasets. From Table I, it is clear that the proposed LS-FLSTSVM-CIL classifier showed the best overall win–tie–loss figures. The proposed LS-FLSTSVM-CIL classifier successfully won in seven datasets and lost in three datasets followed by the RFLSTSVM-CIL, which won in six datasets and lost in three datasets. The TSVM, the LSTSVM, and the FTSVM won in five, four, and two datasets, respectively. Fig. 2 shows the plots of F-measure and G-mean. It is evident in most of the datasets that the proposed LS-FLSTSVM-CIL demonstrated overall superior performance in contrast to other classifiers.

B. Experiments on NDC [40] Datasets

Here, we empirically set the penalty parameters of the baseline models and the proposed model to 10^3, i.e., $c_1 = c_2 = c_3 = c_4 = \mu = 10^3$. Table II gives the performance measures [AUC, Time(s)] of the proposed LS-FLSTSVM-CIL classifier and other classifiers. In NDC-$10k$ and NDC-$20k$, the proposed model achieved 97% and 87% accuracy, respectively, which is comparatively lesser that the other given model. However, the time taken by the proposed model is significantly lower compared to existing models. In NDC-$30k$, the proposed LS-FLSTSVM-CIL model achieved the highest accuracy in lesser training time. As the dataset size increases to 40k and 50k, the proposed LS-FLSTSVM-CIL model achieves competitive performance with less training time. As the sample size crosses 50k, either the existing models run out of memory or the training time of the existing models is worse. However, the proposed LS-FLSTSVM-CIL still gives competitive performance with lesser training time. Thus, it is evident from Table II that the training time of TSVM, LSTSVM, FTSVM, and RFLSTSVM-CIL increases as the size of the dataset increases. As the size of the dataset increases, TSVM, FTSVM, and RFLSTSVM-CIL stopped due to memory issues. However, the proposed LS-FLSTSVM-CIL classifier is faster in contrast to other models and achieved comparable or better performance. Fig. 1 gives the performance of the proposed LS-FLSTSVM-CIL model based on the range of IRs taken in the NDC dataset with 90k sample size. From the figure, it is clear that the performance of the proposed LS-FLSTSVM-CIL model in the large-scale imbalance dataset is almost consistent.

C. Statistical Analysis With the Friedman Test

Here, we evaluate the classification models statistically. We follow the Friedman test [41] for statistical analysis of the classifiers. In this method, each classifier is given a rank corresponding to every data with the best-performing classifier being assigned a lower rank, whereas the worst-performing classifier is assigned a higher rank. Lower the rank of a classifier, the better the classifier it represents. On the ith dataset, suppose that r_i^j is the rank of the jth classifier. According to null hypothesis, all the classifiers are equal, and the corresponding ranks ($\bar{R}_j$) should be equal. If k classifiers are evaluated on N datasets, then the Friedman statistic is given as $\chi_F^2 = \frac{12\ N}{k(k+1)} \left[\sum_j \bar{R}_j^2 - \frac{k(k+1)^2}{4} \right]$, which follows χ_F^2 distribution with $(k-1)$ degrees of freedom, when N and k are large enough. The Friedman statistic is conservative, and hence, $F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2}$ is used, which follows F-distribution with $(k-1)$ and $(k-1)(N-1)$ degrees of freedom. Based on null hypothesis evaluation, we evaluate the significance of classification models.

Based on average ranking given in Table I and after simple calculations, we get $\chi_F^2 = 5.5872$ and $F_F = 1.4149$. F_F is distributed with $(k-1)$ and $(k-1)(N-1)$ degrees of freedom. At $\alpha = 0.05$, the critical value of $F_F(4, 124) = 2.44$; hence, the Friedman test did not show the difference between the classifiers. However, it is evident from Table I that the proposed LS-FLSTSVM-CIL demonstrated lower average rank and higher average accuracy in contrast to baseline classifiers.

VII. Biomedical Applications

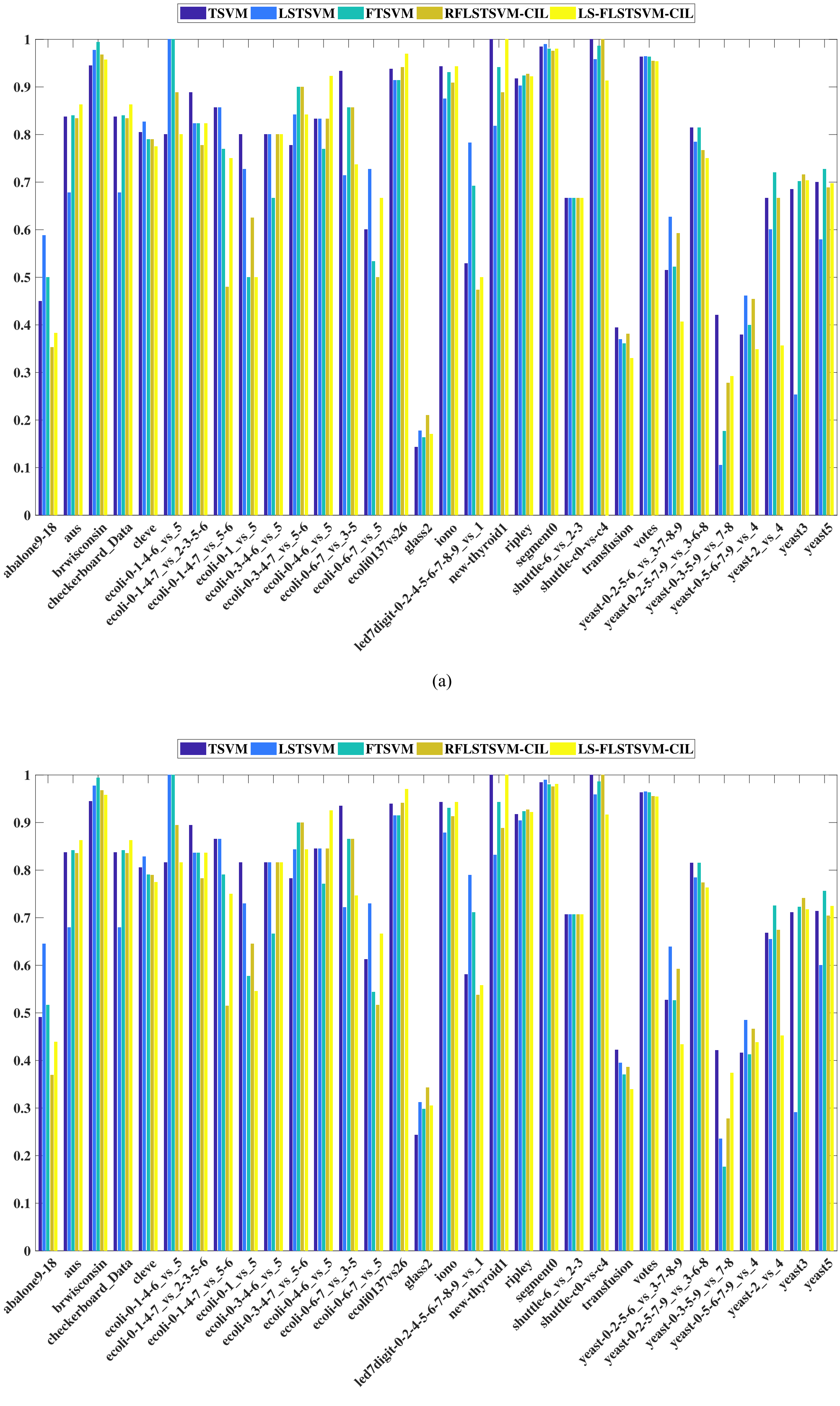
Here, we discuss the potential applications of the proposed LS-FLSTSVM-CIL classifier.

A. Diagnosis of AD

Alzheimer's data used in this study belong to Alzheimer's Disease Neuroimaging Initiative (ADNI) repository (http://adni.loni.usc.edu/). In 2003, Michael W. Weiner, Principal investigator, launched ADNI. The main aim was to evaluate AD via neuroimaging approaches, such as positron emission tomography, magnetic resonance imaging (MRI), and other clinical neuropsychological tests. For more information, visit www.adni-info.org. We follow the ADNI baseline dataset [42], [43] and downloaded 817 structural MRI images. For the volume-based morphometry of the images, we used the FreeSurfer's recon-all pipeline (version 6.0.1) [44], [45]. Out of 817 images, four images fail to

TABLE I
PERFORMANCE OF TSVM, LSTSVM, FTSVM, RFLSTSVM-CIL, AND LS-FLSTSVM-CIL

Datasets	TSVM (AUC, Time(s)) $(c_1 = c_2, \mu)$	LSTSVM (AUC, Time(s)) $(c_1 = c_2, \mu)$	FTSVM (AUC, Time(s)) $(c_1 = c_2, \mu)$	RFLSTSVM-CIL (AUC, Time(s) $(c_0, c_1 = c_2, \mu)$	Proposed LS-FLSTSVM-CIL (AUC, Time(s)) $(c_0, c_1 = c_2, c_3 = c_4, \mu)$
abalone9-18	$(0.8295, 0.1247)$ $(10^2, 1)$	$(0.7083, 0.0331)$ $(1, 0.1)$	$(0.8046, 0.1078)$ $(10^{-2}, 1)$	$(0.7117, 0.0446)$ $(2, 0.1, 0.1)$	$(0.8128, 0.0338)$ $(0.5, 10^5, 10^4, 0.1)$
ecoli-0-3-4-6_vs_5	$(0.8333, 0.0133)$ $(10^{-2}, 10^4)$	$(0.8333, 0.0026)$ $(0.1, 10^4)$	$(0.825, 0.0127)$ $(1, 10^4)$	$(0.8333, 0.0042)$ $(1, 1, 10^4)$	$(0.8333, 0.0053)$ $(0.5, 1, 1, 10^3)$
ecoli-0-3-4-7_vs_5-6	$(0.8428, 0.0221)$ $(1, 10^3)$	$(0.8928, 0.0035)$ $(1, 10^4)$	$(0.9428, 0.019)$ $(1, 10^4)$	$(0.9428, 0.0056)$ $(0.5, 1, 10^4)$	$(0.8928, 0.0065)$ $(1.5, 1, 1, 10^3)$
aus	$(0.8516, 0.0702)$ $(0.1, 10^2)$	$(0.7191, 0.0234)$ $(10^{-4}, 10^4)$	$(0.8512, 0.0632)$ $(10^{-5}, 10^4)$	$(0.8458, 0.0262)$ $(0.5, 1, 10^4)$	$(0.8753, 0.0218)$ $(2, 10, 0.1, 10^4)$
brwisconsin	$(0.9685, 0.0733)$ $(10^{-3}, 10^2)$	$(0.9858, 0.0255)$ $(10^{-2}, 10)$	$(0.9938, 0.072)$ $(10^{-5}, 10^2)$	$(0.9907, 0.0306)$ $(0.5, 10^{-2}, 10^5)$	$(0.9876, 0.0214)$ $(0.5, 10^4, 10^4, 10^4)$
ecoli-0-1-4-7_vs_2-3-5-6	$(0.9, 0.0264)$ $(1, 10^4)$	$(0.85, 0.006)$ $(10, 10^3)$	$(0.85, 0.0266)$ $(0.1, 10^4)$	$(0.8446, 0.0082)$ $(1, 10^{-2}, 10^4)$	$(0.85, 0.0073)$ $(2.5, 0.1, 1, 10^3)$
ecoli-0-1-4-7_vs_5-6	$(0.875, 0.0248)$ $(10, 10^5)$	$(0.875, 0.0059)$ $(1, 10^4)$	$(0.8125, 0.0256)$ $(1, 10^4)$	$(0.8159, 0.008)$ $(1.5, 10^{-2}, 10^5)$	$(0.8642, 0.0072)$ $(2, 10^5, 10^4, 10^3)$
checkerboard_Data	$(0.8516, 0.0697)$ $(0.1, 10^2)$	$(0.7191, 0.0228)$ $(10^{-4}, 10^4)$	$(0.8512, 0.063)$ $(10^{-5}, 10^4)$	$(0.8458, 0.0269)$ $(0.5, 1, 10^4)$	$(0.8753, 0.0219)$ $(2, 10, 0.1, 10^4)$
cleve	$(0.8287, 0.0191)$ $(10^{-5}, 10^4)$	$(0.8483, 0.0046)$ $(10^{-3}, 10^3)$	$(0.8162, 0.0168)$ $(10^{-5}, 10^4)$	$(0.8118, 0.0069)$ $(2.5, 0.1, 10^4)$	$(0.7993, 0.0069)$ $(0.5, 10^2, 1, 10^4)$
ecoli-0-1_vs_5	$(0.8333, 0.0173)$ $(1, 10^5)$	$(0.8259, 0.0032)$ $(1, 10^4)$	$(0.6667, 0.0181)$ $(0.1, 10^5)$	$(0.8794, 0.0055)$ $(1.5, 10^{-2}, 10^4)$	$(0.8495, 0.0051)$ $(0.5, 10^{-3}, 0.1, 10^3)$
ecoli-0-1-4-6_vs_5	$(0.9877, 0.0221)$ $(10, 10^5)$	$(1, 0.0044)$ $(1, 10^4)$	$(1, 0.0218)$ $(10^{-3}, 10^3)$	$(0.9938, 0.0062)$ $(1.5, 10^{-5}, 10^4)$	$(0.9877, 0.006)$ $(2, 10^{-2}, 1, 10^3)$
ecoli-0-4-6_vs_5	$(0.8571, 0.0125)$ $(0.1, 10^4)$	$(0.8571, 0.0023)$ $(1, 10^4)$	$(0.8481, 0.0142)$ $(10^{-2}, 10^4)$	$(0.8571, 0.0043)$ $(2, 10^{-5}, 10^3)$	$(0.9286, 0.0053)$ $(0.5, 1, 1, 10^3)$
ecoli0137vs26	$(0.9412, 0.0294)$ $(1, 1)$	$(0.9578, 0.0048)$ $(1, 0.1)$	$(0.9578, 0.0204)$ $(10^{-5}, 10^2)$	$(0.9642, 0.0068)$ $(0.5, 0.1, 10^2)$	$(0.9706, 0.0069)$ $(1, 1, 1, 0.1)$
glass2	$(0.5927, 0.0163)$ $(10^2, 10^5)$	$(0.7016, 0.003)$ $(10^2, 10^4)$	$(0.6694, 0.0145)$ $(10^{-2}, 10^3)$	$(0.7581, 0.0047)$ $(2, 10^{-2}, 10^2)$	$(0.6855, 0.0052)$ $(2.5, 10^{-2}, 1, 1)$
iono	$(0.9575, 0.0254)$ $(10^{-3}, 1)$	$(0.8931, 0.0062)$ $(0.1, 10)$	$(0.8938, 0.0196)$ $(10^{-2}, 10^2)$	$(0.9514, 0.0087)$ $(0.5, 10^{-3}, 1)$	$(0.9575, 0.0084)$ $(1.5, 10^3, 10, 1)$
led7digit-0-2-4-5-6-7-8-9_vs_1	$(0.8895, 0.0529)$ $(10^2, 10^3)$	$(0.9339, 0.0095)$ $(0.1, 1)$	$(0.9218, 0.0427)$ $(10, 10^3)$	$(0.8734, 0.0116)$ $(1.5, 1, 10^3)$	$(0.8815, 0.01)$ $(2, 1, 10, 1)$
new-thyroid1	$(1, 0.0124)$ $(1, 10^3)$	$(0.9649, 0.0027)$ $(100, 10^3)$	$(0.9444, 0.0131)$ $(10^{-5}, 10^3)$	$(0.9357, 0.005)$ $(0.5, 0.1, 10^5)$	$(1, 0.0054)$ $(0.5, 10^5, 10^4, 10^2)$
ripley	$(0.9178, 0.2179)$ $(10^{-2}, 1)$	$(0.9079, 0.0871)$ $(10^{-5}, 0.01)$	$(0.923, 0.2384)$ $(0.1, 0.1)$	$(0.9254, 0.1149)$ $(2, 1, 0.1)$	$(0.9201, 0.0775)$ $(0.5, 10^4, 10^4, 0.1)$
ecoli-0-6-7_vs_3-5	$(0.9375, 0.0189)$ $(1, 10^4)$	$(0.8042, 0.003)$ $(0.1, 10^3)$	$(0.875, 0.0156)$ $(1, 10^4)$	$(0.875, 0.0048)$ $(1, 10^{-5}, 10^4)$	$(0.9042, 0.0052)$ $(1.5, 10^2, 10, 10^4)$
ecoli-0-6-7_vs_5	$(0.7418, 0.0174)$ $(10^{-2}, 10^3)$	$(0.8251, 0.0031)$ $(1, 10^4)$	$(0.7923, 0.0153)$ $(10^{-5}, 10^4)$	$(0.7842, 0.0049)$ $(2.5, 10^{-5}, 10^4)$	$(0.8169, 0.0053)$ $(1, 10^5, 10^5, 10^3)$
segment0	$(0.9892, 1.0173)$ $(10, 10^4)$	$(0.99, 0.3218)$ $(10^{-2}, 10^3)$	$(0.9842, 0.9464)$ $(10^{-2}, 10^3)$	$(0.9916, 0.3832)$ $(1, 0.1, 10^4)$	$(0.9966, 0.2497)$ $(0.5, 10, 1, 10^2)$
shuttle-6_vs_2-3	$(0.75, 0.0189)$ $(10^{-5}, 10^5)$	$(0.75, 0.0035)$ $(10^{-4}, 10^5)$	$(0.75, 0.017)$ $(10^{-5}, 10^5)$	$(0.75, 0.0055)$ $(0.5, 10^{-5}, 10^5)$	$(0.75, 0.0051)$ $(0.5, 0.1, 0.1, 10^5)$
shuttle-c0-vs-c4	$(1, 0.8593)$ $(10^{-2}, 10^5)$	$(0.9595, 0.2012)$ $(0.1, 10^3)$	$(0.9865, 0.5173)$ $(1, 10^5)$	$(1, 0.2474)$ $(2.5, 0.1, 10^4)$	$(0.9932, 0.1594)$ $(0.5, 0.1, 1, 10^5)$
transfusion	$(0.6255, 0.1066)$ $(10, 10^5)$	$(0.6116, 0.0338)$ $(1, 10^3)$	$(0.6065, 0.0937)$ $(0.1, 10^5)$	$(0.6146, 0.0442)$ $(0.5, 0.1, 10^4)$	$(0.5731, 0.0307)$ $(1.5, 10^5, 10^4, 10^3)$
votes	$(0.9687, 0.0293)$ $(1, 10^2)$	$(0.9744, 0.0081)$ $(1, 10^2)$	$(0.9687, 0.0274)$ $(10^4, 10^3)$	$(0.9651, 0.0105)$ $(1.5, 0.1, 10^2)$	$(0.9623, 0.0097)$ $(0.5, 10^2, 1, 10)$
yeast-0-2-5-6_vs_3-7-8-9	$(0.6954, 0.2196)$ $(10, 10^2)$	$(0.7511, 0.062)$ $(10^2, 0.01)$	$(0.7468, 0.1838)$ $(0.1, 10^2)$	$(0.7677, 0.0827)$ $(1, 1, 10^2)$	$(0.7022, 0.0582)$ $(1, 10^2, 10^4, 0.1)$
yeast-0-2-5-7-9_vs_3-6-8	$(0.9122, 0.2024)$ $(10, 1)$	$(0.8756, 0.0609)$ $(10^{-3}, 0.01)$	$(0.9122, 0.192)$ $(10, 1)$	$(0.9225, 0.0798)$ $(1, 0.1, 1)$	$(0.9363, 0.0565)$ $(2, 10, 10, 1)$
yeast-0-3-5-9_vs_7-8	$(0.6778, 0.0517)$ $(10^2, 10^5)$	$(0.5278, 0.012)$ $(0.1, 0.1)$	$(0.5352, 0.0534)$ $(1, 0.1)$	$(0.5907, 0.0138)$ $(2, 10^{-2}, 0.1)$	$(0.6519, 0.012)$ $(2.5, 10^4, 10^3, 0.1)$
yeast-0-5-6-7-9_vs_4	$(0.7186, 0.0644)$ $(100, 10^5)$	$(0.666, 0.0154)$ $(1, 0.1)$	$(0.6983, 0.0506)$ $(10^{-5}, 10^2)$	$(0.7347, 0.0177)$ $(2, 1, 10^2)$	$(0.7524, 0.0158)$ $(2, 10^2, 10^2, 0.1)$
yeast-2_vs_4	$(0.836, 0.0522)$ $(10^2, 1)$	$(0.7143, 0.0122)$ $(1, 0.1)$	$(0.8144, 0.0596)$ $(0.1, 1)$	$(0.8647, 0.0142)$ $(0.5, 0.1, 0.1)$	$(0.8023, 0.0123)$ $(2, 10^4, 10^3, 0.1)$
yeast3	$(0.9012, 0.4921)$ $(10^2, 10^2)$	$(0.5702, 0.1324)$ $(10^{-4}, 10^4)$	$(0.9013, 0.322)$ $(10^{-5}, 10^2)$	$(0.9218, 0.1669)$ $(2.5, 10^{-2}, 1)$	$(0.8861, 0.1039)$ $(1.5, 10^3, 1, 10^2)$
yeast5	$(0.9198, 0.6107)$ $(10^3, 10^4)$	$(0.7256, 0.1413)$ $(1, 0.01)$	$(0.9787, 0.4508)$ $(10^{-2}, 10^2)$	$(0.9186, 0.1736)$ $(1.5, 10^{-3}, 1)$	$(0.9567, 0.1136)$ $(2, 10^4, 10^3, 1)$
Average AUC	0.8572	0.8194	0.8476	0.8588	**0.8642**
Average Rank	2.8438	3.4844	3.25	2.7813	**2.6406**
Overall Win-Tie-Loss	$5-1-6$	$4-1-13$	$2-1-5$	$6-1-3$	$\mathbf{7-1-3}$



(a)

(b)

Fig. 2. Performance measures corresponding to different datasets with the baseline models and the proposed LS-FLSTSVM-CIL model. (a) F-measure analysis. (b) G-mean analysis.

TABLE II
COMPARISON OF AUC AND TRAINING TIME (SECONDS) OF THE CLASSIFICATION MODELS ON NDC [40] DATASETS

Datasets	(Train, Test)	TSVM (AUC,Time(s))	LSTSVM (AUC,Time(s))	FTSVM (AUC,Time(s))	RFLSTSVM-CIL (AUC,Time(s))	LS-FLSTSVM-CIL (AUC,Time(s))
NDC-10k	$(10000, 100)$	$(1, 66.22)$	$(0.99, 33.2)$	$(1, 64.59)$	$(0.97, 34.25)$	$(0.97, 12.63)$
NDC-20k	$(20000, 200)$	$(1, 470.89)$	$(0.89, 227.21)$	$(1, 435.33)$	$(0.94, 234.83)$	$(0.87, 39.39)$
NDC-30k	$(30000, 300)$	$(0.99, 1551.45)$	$(0.77, 710.04)$	$(0.99, 1459.24)$	$(0.91, 736.62)$	$(1, 87.66)$
NDC-40k	$(40000, 400)$	$(1, 3653.39)$	$(0.96, 1664.56)$	$(1, 3591.07)$	$(0.9, 1675.39)$	$(0.99, 167.95)$
NDC-50k	$(50000, 500)$	$(0.99, 22804.3)$	$(0.87, 3099.81)$	$(0.99, 7903.11)$	$(0.76, 3827.44)$	$(0.99, 279.37)$
NDC-60k	$(60000, 600)$	*	$(0.52, 5447.58)$	*	*	$(0.99, 379.19)$
NDC-70k	$(70000, 700)$	*	$(0.66, 41173.9)$	*	*	$(0.61, 544.87)$
NDC-80k	$(80000, 800)$	*	#	*	*	$(0.99, 834.59)$
NDC-90k	$(90000, 900)$	*	#	*	*	$(0.89, 1485.5)$

*Denotes out of memory.$^{\#}$Experiments stopped due to large training time.

TABLE III
APPLICATIONS ON BIOMEDICAL DATA

Datasets (Train, Test)	TSVM AUC (F-Measure, G-Mean)	LSTSVM AUC (F-Measure, G-Mean)	FTSVM AUC (F-Measure, G-Mean)	RFLSTSVM-CIL AUC (F-Measure, G-Mean)	LS-FLSTSVM-CIL AUC (F-Measure, G-Mean)
CN_vs_MCI subjects	0.6192	0.5906	0.6197	$\mathbf{0.6748}$	0.616
$437 \times 91, 189 \times 91)$	$(0.6445, 0.6516)$	$(0.4182, 0.4304)$	$(0.6844, 0.6864)$	$(0.6127, 0.6272)$	$(0.5455, 0.5541)$
MCI_vs_AD subjects	0.6299	0.5972	0.6244	0.6597	$\mathbf{0.6757}$
$408 \times 91, 177 \times 91)$	$(0.5038, 0.5132)$	$(0.3797, 0.4041)$	$(0.4874, 0.4903)$	$(0.5478, 0.5792)$	$(0.5564, 0.5682)$
AN_vs_DC subjects	$\mathbf{0.8901}$	0.814	0.8511	0.822	0.8848
$218 \times 768, 96 \times 768)$	$(0.8649, 0.8649)$	$(0.7714, 0.7727)$	$(0.8158, 0.8161)$	$(0.7789, 0.7987)$	$(0.8471, 0.8542)$
AN_vs_LC subjects	0.5607	0.4708	$\mathbf{0.5746}$	0.4788	0.5446
$169 \times 768, 74 \times 768)$	$(0.4211, 0.4466)$	$(0.0909, 0.1325)$	$(0.4286, 0.4588)$	$(0.6042, 0.6177)$	$(0.6733, 0.6949)$
AN_vs_MC subjects	$\mathbf{0.611}$	0.6064	0.5932	0.6413	0.5881
$191 \times 768, 84 \times 768)$	$(0.5556, 0.5564)$	$(0.5294, 0.5331)$	$(0.5075, 0.5121)$	$(0.6972, 0.7316)$	$(0.617, 0.6287)$
AN_vs_PC subjects	$\mathbf{0.7457}$	0.5768	0.6799	0.6369	0.7308
$169 \times 768, 75 \times 768)$	$(0.7654, 0.7669)$	$(0.4286, 0.4588)$	$(0.6842, 0.6842)$	$(0.7097, 0.7218)$	$(0.7778, 0.7874)$
FA_vs_DC subjects	0.845	0.7633	$\mathbf{0.86}$	0.8517	0.81
$310 \times 768, 135 \times 768)$	$(0.8571, 0.8573)$	$(0.7304, 0.7311)$	$(0.8707, 0.8709)$	$(0.8361, 0.8362)$	$(0.7863, 0.7866)$
FA_vs_LC subjects	0.6168	0.6236	0.6306	0.6153	$\mathbf{0.6672}$
$260 \times 768, 114 \times 768)$	$(0.7532, 0.7543)$	$(0.4932, 0.4969)$	$(0.6718, 0.6762)$	$(0.4935, 0.4946)$	$(0.6, 0.61)$
FA_vs_MC subjects	0.5285	0.5422	0.5557	$\mathbf{0.5558}$	0.5117
$283 \times 768, 123 \times 768)$	$(0.6164, 0.6165)$	$(0.466, 0.4666)$	$(0.6345, 0.6346)$	$(0.4906, 0.492)$	$(0.4381, 0.4391)$
FA_vs_PC subjects	0.5337	0.5221	0.5765	0.6574	$\mathbf{0.7152}$
$261 \times 768, 114 \times 768)$	$(0.6928, 0.6935)$	$(0.5369, 0.6011)$	$(0.6569, 0.6584)$	$(0.5981, 0.6152)$	$(0.6452, 0.6497)$
PT_vs_DC subjects	$\mathbf{0.8977}$	$\mathbf{0.8977}$	0.8722	0.8807	0.8935
$225 \times 768, 98 \times 768)$	$(0.88, 0.8807)$	$(0.88, 0.8807)$	$(0.8462, 0.8462)$	$(0.8571, 0.8572)$	$(0.8718, 0.8718)$
PT_vs_LC subjects	0.4692	0.4658	0.5024	0.56	$\mathbf{0.5657}$
$175 \times 768, 77 \times 768)$	$(0.3226, 0.3408)$	$(0.3636, 0.3748)$	$(0.4348, 0.4427)$	$(0.5641, 0.5648)$	$(0.6863, 0.6999)$
PT_vs_MC subjects	0.5237	0.5686	0.596	$\mathbf{0.621}$	0.6059
$197 \times 768, 87 \times 768)$	$(0.2642, 0.307)$	$(0.4375, 0.4518)$	$(0.5679, 0.5679)$	$(0.6024, 0.6028)$	$(0.6538, 0.672)$
PT_vs_PC subjects	0.607	0.5843	0.6386	0.6264	$\mathbf{0.7219}$
$176 \times 768, 77 \times 768)$	$(0.6429, 0.643)$	$(0.6, 0.6002)$	$(0.7097, 0.7147)$	$(0.6957, 0.6998)$	$(0.7586, 0.7599)$
TA_vs_DC subjects	0.7371	0.7553	0.7566	0.7553	$\mathbf{0.7686}$
$235 \times 768, 103 \times 768)$	$(0.6829, 0.6931)$	$(0.7, 0.7144)$	$(0.7073, 0.7179)$	$(0.7, 0.7144)$	$(0.76, 0.7606)$
TA_vs_LC subjects	$\mathbf{0.7782}$	0.6409	0.7365	0.6422	0.6979
$185 \times 768, 82 \times 768)$	$(0.8085, 0.8087)$	$(0.5789, 0.6001)$	$(0.7556, 0.7572)$	$(0.6882, 0.6885)$	$(0.7963, 0.8013)$
TA_vs_MC subjects	0.4643	0.4966	$\mathbf{0.6164}$	0.5524	0.606
$208 \times 768, 91 \times 768)$	$(0.2899, 0.3074)$	$(0.3784, 0.3901)$	$(0.5783, 0.5817)$	$(0.3881, 0.4183)$	$(0.5385, 0.5474)$
TA_vs_PC subjects	0.6593	0.5441	0.565	0.5607	$\mathbf{0.7243}$
$186 \times 768, 82 \times 768)$	$(0.6588, 0.6644)$	$(0.5581, 0.562)$	$(0.5909, 0.5934)$	$(0.6, 0.6013)$	$(0.7143, 0.7217)$
Average-AUC	0.651	0.6145	0.6583	0.6551	$\mathbf{0.6849}$
Average-Rank	3.0278	4.2778	2.5556	2.8056	$\mathbf{2.3333}$
Overall Win-Tie-Loss	$3-0-3$	$0-0-11$	$3-0-1$	$4-0-1$	$\mathbf{7-0-2}$

process via the FreeSurfer pipeline. Thus, we use mild cognitive impairment (MCI), AD, and control normal (CN) subjects with 228, 398, and 187 images, respectively. Table III (first two rows) gives the classification performance on CN versus MCI and MCI versus AD subjects. It is evident that the proposed LS-FLSTSVM-CIL classifier showed superior performance on MCI versus AD subjects, which are hard to distinguish.

B. Diagnosis of Breast Cancer Disease

The BreakHis histopathological breast cancer image dataset [46] is used for evaluation. Total 1240 scans are used with 400× magnification. The dataset involves benign and malignant categories. Benign category is subdivided into phyllodes tumor (PT), adenosis (AN), tubular adenoma (TA), and fibroadenoma

TABLE IV
PERFORMANCE OF CLASSIFICATION MODELS ON CARCINOM, LUNG, AND TDT2 DATASETS

Domain	Datasets	Details $Samples \times Features \times Class$	TSVM (AUC,Time(s))	LSTSVM (AUC,Time(s))	FTSVM (AUC,Time(s))	RFLSTSVM-CIL (AUC,Time(s))	LS-FLSTSVM-CIL (AUC,Time(s))
Biology	Carcinom	$174 \times 9182 \times 11$	$(\mathbf{0.87}, 0.38)$	$(0.83, 0.36)$	$(\mathbf{0.87}, 0.78)$	$(0.83, 0.77)$	$(0.83, 1.22)$
Biology	Lung	$203 \times 3312 \times 5$	$(0.87, 0.13)$	$(0.61, 0.1)$	$(0.87, 0.2)$	$(0.61, 0.2)$	$(\mathbf{0.89}, 0.32)$
Text	TDT2	$9394 \times 1000 \times 30$	$(\mathbf{0.93}, 1683.17)$	$(0.66, 467.83)$	$(\mathbf{0.93}, 73819.2)$	$(0.71, 924.55)$	$(0.91, 854.82)$

(FA) having 115, 106, 130, and 237 images, respectively. Malignant category is subdivided into mucinous carcinoma (MC), ductal carcinoma (DC), papillary carcinoma (PC), and lobular carcinoma (LC), which contain 169, 208, 138, and 137 images, respectively. The images are converted into gray-level images, and Daubechies-4 wavelet up to three levels of decomposition is applied on each image [47]. The concatenation of approximation and detail coefficients is used as the feature vector. Table III (from third row onwards), it is evident that the proposed LS-FLSTSVM-CIL classifier demonstrated superior performance in contrast to the existing models in most of the cases in breast cancer subject classification. Thus, the proposed model can be used in clinical setting for the automation of breast cancer classification.

VIII. DISCUSSION OF RESULTS

Here, we discuss the classification performance of the classifiers on UCI [38], KEEL [39], NDC [40], and biomedical [42], [43], [46] datasets.

The models are evaluated in terms of AUC on UCI and KEEL datasets (given in Table I). One can see that the TSVM achieved 85.72% AUC, the LSTSVM achieved 81.94% AUC, the FTSVM achieved 84.76% AUC, RFLSTSVM-CIL achieved 85.88% AUC, and the proposed LS-FLSTSVM-CIL model achieved 86.42% AUC. In addition, with respect to average rank, the proposed LS-FLSTSVM-CIL classifier showed best figures with 2.6406 followed by the RFLSTSVM-CIL model with 2.7813, and TSVM, FTSVM, and LSTSVM models with 2.8438, 3.25, and 3.4844, respectively. Furthermore, the overall win–tie–loss performance measure of the classifiers is evaluated. The proposed LS-FLSTSVM-CIL classifier achieved the highest win performance over seven datasets. The RFLSTSVM-CIL and the proposed LS-FLSTSVM-CIL model achieved the lowest overall losses over three datasets. The lowest overall wins resulted in the FTSVM model with two datasets. The further evaluation of the models in terms of F-measure and G-mean (see Fig. 2) demonstrates the efficacy of the proposed LS-FLSTSVM-CIL classifier over the existing baseline classifiers.

In order to demonstrate the effectiveness of the proposed LS-FLSTSVM-CIL classifier on large-scale data, we used NDC datasets. Table II gives the performance of the models in terms of AUC and training time. It is evident that the proposed LS-FLSTSVM-CIL classifier is competitive with respect to the baseline classifiers. Moreover, the time taken for training the proposed LS-FLSTSVM-CIL classifier is significantly lower in contrast to baseline classifiers. In addition, as the sample size increases, either the baseline models go out of memory or the training time is very high. This proves the effectiveness of the proposed LS-FLSTSVM-CIL classifier in large-scale problems.

To demonstrate the applicability of the proposed LS-FLSTSVM-CIL classifier in real-world applications, we used the proposed LS-FLSTSVM-CIL model for the diagnosis of AD and breast cancer disease. Table III presents the performance of classifiers in the biomedical domain. Table III gives the performance in terms of AUC, F-measure, and G-mean. One can see that the proposed LS-FLSTSVM-CIL model showed better performance in the classification of the MCI versus AD subjects. In addition, with respect to accuracy, the proposed LS-FLSTSVM-CIL classifier is mostly superior in subcategories of breast cancer disease. The proposed LS-FLSTSVM-CIL model demonstrated the highest average AUC and the lowest average rank. Thus, the practical application of the proposed LS-FLSTSVM-CIL classifier in the biomedical domain shows its feasibility in real-world scenarios. The performance of the models on datasets from Biology and Text domains is shown in Table IV. One can see that the performance of the proposed LS-FLSTSVM-CIL model is competitive to the existing models with lower training time especially in the TDT2 dataset with 10 000 samples.

Thus, the proposed LS-FLSTSVM-CIL is competitive or better in terms of AUC, and the training time is lower in contrast to the existing classifiers, which proves the effectiveness of the proposed LS-FLSTSVM-CIL classifier.

IX. CONCLUSION

In this article, we proposed LS-FLSTSVM-CIL. The proposed LS-FLSTSVM-CIL implemented the structural risk minimization principle, which led to the better generalization of the proposed LS-FLSTSVM-CIL. In addition, the Wolfe dual of the proposed LS-FLSTSVM-CIL formulation involved positive-definite matrices. The major advantage of the proposed LS-FLSTSVM-CIL classifier is that it avoids the calculation of matrix inverse, which makes it feasible for large-scale problems. To make it more efficient for large-scale problems, we solved the objective function of the proposed LS-FLSTSVM-CIL classifier via the iterative procedure known as SMO. Furthermore, the kernel trick can be directly applied in the proposed LS-FLSTSVM-CIL for the nonlinear case. The proposed LS-FLSTSVM-CIL model gives the appropriate weights to the samples of the majority class based on the sample distance from the centroids of the positive/negative class and the IR of the datasets. Unlike the standard TSVM, wherein each data point is treated equally for the design of decision boundary, the proposed LS-FLSTSVM-CIL model weights the samples based on their location and IR of the data. This results in reducing the effect of outliers and also reduces the effect of bias toward the majority class samples. The performance of the proposed LS-FLSTSVM-CIL on UCI and KEEL datasets demonstrated the ability of the proposed LS-FLSTSVM-CIL classifier. The proposed LS-FLSTSVM-CIL is scalable to large-scale datasets, which is shown by its effectiveness in the large-scale NDC datasets. To demonstrate the practical applications of the proposed LS-FLSTSVM-CIL classifier, we evaluated it for the diagnosis of breast cancer disease and AD. Experimental evaluation demonstrate the efficacy of the proposed LS-FLSTSVM-CIL classifier. It is perceptible that the proposed LS-FLSTSVM-CIL classifier is efficient both in terms of time and accuracy. The proposed LS-FLSTSVM-CIL classifier has some limitations, which involve the computation overhead involved in the optimization of multiple hyperparameters that need to be tuned optimally for better generalization. In addition, the proposed classifier is formulated for binary class problems. One can also think to extend the formulation to multiclass problems. In future, one can explore more efficient techniques for parameter optimization. In addition, one can focus on developing more robust fuzzy functions to deal with the issues of noise and outliers more robustly. The code of the proposed LS-FLSTSVM-CIL will be available on the author's homepage.[1]

Acknowledgment

The authors are grateful to the Indian Institute of Technology Indore for the facilities and support provided for this work. Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset was used in this study (adni.loni.usc.edu). The design and implementation of ADNI was done by ADNI investigators, but did not participate either in analysis or in writing of this article. A comprehensive list of ADNI investigators is available at.[2] The Canadian Institutes of Health Research provided funds to support ADNI clinical sites in Canada. The Foundation for the National Institutes of Health[3] provided private sector contributions. The grantee organization was the Northern California Institute for Research and Education, and the study was coordinated by the Alzheimer's Therapeutic Research Institute, University of Southern California. The dissemination of ADNI data was carried out by the Laboratory for Neuro Imaging, University of Southern California.

References

[1] V. Vapnik, *Nature of Statistical Learning Theory* (ser. Statistics for Engineering and Information Science). New York, NY, USA: Springer, 2000.

[2] B. R. Froz, A. O. de Carvalho Filho, A. C. Silva, A. C. de Paiva, R. A. Nunes, and M. Gattass, "Lung nodule classification using artificial crawlers, directional texture and support vector machine," *Expert Syst. Appl.*, vol. 69, pp. 176–188, 2017.

[3] L. Zago *et al.*, "Predicting hemispheric dominance for language production in healthy individuals using support vector machine," *Human Brain Mapping*, vol. 38, no. 12, pp. 5871–5889, 2017.

[4] O. L. Mangasarian and E. W. Wild, "Multisurface proximal support vector machine classification via generalized eigenvalues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 69–74, Jan. 2006.

[5] Jayadeva, R. Khemchandani, and S. Chandra, "Twin support vector machines for pattern classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5, pp. 905–910, May 2007.

[6] Y.-H. Shao, C.-H. Zhang, X.-B. Wang, and N.-Y. Deng, "Improvements on twin support vector machines," *IEEE Trans. Neural Netw.*, vol. 22, no. 6, pp. 962–968, Jun. 2011.

[7] Y. Tian, Z. Qi, X. Ju, Y. Shi, and X. Liu, "Nonparallel support vector machines for pattern classification," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1067–1079, Jul. 2014.

[8] H. Wang, Y. Shi, L. Niu, and Y. Tian, "Nonparallel support vector ordinal regression," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3306–3317, Oct. 2017.

[9] X. Miao, Y. Liu, H. Zhao, and C. Li, "Distributed online one-class support vector machine for anomaly detection over networks," *IEEE Trans. Cybern.*, vol. 49, no. 4, pp. 1475–1488, Apr. 2019.

[10] S. Sun, X. Xie, and C. Dong, "Multiview learning with generalized eigenvalue proximal support vector machines," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 688–697, Feb. 2019.

[11] C.-F. Lin and S.-D. Wang, "Fuzzy support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 464–471, Mar. 2002.

[12] D. Tsujinishi and S. Abe, "Fuzzy least squares support vector machines for multiclass problems," *Neural Netw.*, vol. 16, nos. 5/6, pp. 785–792, 2003.

[13] Y. Wang, S. Wang, and K. K. Lai, "A new fuzzy support vector machine to evaluate credit risk," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 6, pp. 820–831, Dec. 2006.

[14] S. Balasundaram and M. Tanveer, "On proximal bilateral-weighted fuzzy support vector machine classifiers," *Int. J. Adv. Intell. Paradigms*, vol. 4, nos. 3/4, pp. 199–210, 2012.

[15] S. Rezvani, X. Wang, and F. Pourpanah, "Intuitionistic fuzzy twin support vector machines," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 11, pp. 2140–2151, Nov. 2019.

[16] S. Chen, J. Cao, Z. Huang, and C. Shen, "Entropy-based fuzzy twin bounded support vector machine for binary classification," *IEEE Access*, vol. 7, pp. 86 555–86569, 2019.

[17] P.-Y. Hao, "Asymmetric possibility and necessity regression by twin support vector networks," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 10, pp. 3028–3042, Oct. 2021.

[18] Q. Wu, H. Zhang, R. Jing, and Z. Wang, "A class of fuzzy smooth piecewise twin support vector machine," in *Proc. IEEE 14th Int. Conf. Comput. Intell. Secur.*, 2018, pp. 382–385.

[19] U. Gupta, D. Gupta, and M. Prasad, "Kernel target alignment based fuzzy least square twin bounded support vector machine," in *Proc. IEEE Symp. Ser. Comput. Intell.*, 2018, pp. 228–235.

[20] M. A. Ganaie, M. Tanveer, and P. N. Suganthan, "Regularized robust fuzzy least squares twin support vector machine for class imbalance learning," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2020, pp. 1–8.

[21] W. An and M. Liang, "Fuzzy support vector machine based on within-class scatter for classification problems with outliers or noises," *Neurocomputing*, vol. 110, pp. 101–110, 2013.

[22] R. Wang, X. Zhang, and W. Cao, "Clifford fuzzy support vector machines for classification," *Adv. Appl. Clifford Algebras*, vol. 26, no. 2, pp. 825–846, 2016.

[23] M. Tanveer, T. Rajani, R. Rastogi, Y. Shao, and M. A. Ganaie, "Comprehensive review on twin support vector machines," *Ann. Oper. Res.*, 2022, doi: 10.1007/s10479-022-04575-w..

[24] X. Hua and S. Ding, "Weighted least squares projection twin support vector machines with local information," *Neurocomputing*, vol. 160, pp. 228–237, 2015.

[25] J. S. Sartakhti, H. Afrabandpey, and N. Ghadiri, "Fuzzy least squares twin support vector machines," *Eng. Appl. Artif. Intell.*, vol. 85, pp. 402–409, 2019.

[26] T. Razzaghi, O. Roderick, I. Safro, and N. Marko, "Multilevel weighted support vector machine for classification on healthcare data with missing values," *PLoS One*, vol. 11, no. 5, 2016, Art. no. e0155119.

[27] H.-L. Dai, "Class imbalance learning via a fuzzy total margin based support vector machine," *Appl. Soft Comput.*, vol. 31, pp. 172–184, 2015.

[28] D. Tomar, S. Singhal, and S. Agarwal, "Weighted least square twin support vector machine for imbalanced dataset," *Int. J. Database Theory Appl.*, vol. 7, no. 2, pp. 25–36, 2014.

[29] Y. Xu, "Maximum margin of twin spheres support vector machine for imbalanced data classification," *IEEE Trans. Cybern.*, vol. 47, no. 6, pp. 1540–1550, Jun. 2017.

[1] [Online]. Available: https://github.com/mtanveer1/

[2] [Online]. Available: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf

[3] [Online]. Available: www.fnih.org

[30] S.-G. Chen and X.-J. Wu, "A new fuzzy twin support vector machine for pattern classification," *Int. J. Mach. Learn. Cybern.*, vol. 9, no. 9, pp. 1553–1564, 2018.

[31] B. Richhariya and M. Tanveer, "A robust fuzzy least squares twin support vector machine for class imbalance learning," *Appl. Soft Comput.*, vol. 71, pp. 418–432, 2018.

[32] M. A. Kumar and M. Gopal, "Least squares twin support vector machines for pattern classification," *Expert Syst. Appl.*, vol. 36, no. 4, pp. 7535–7543, 2009.

[33] R. Batuwita and V. Palade, "FSVM-CIL: Fuzzy support vector machines for class imbalance learning," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 3, pp. 558–571, Jun. 2010.

[34] M. Grant, S. Boyd, and Y. Ye, "CVX users' guide," Tech. Rep. Build 711, Univ. Cambridge, 2009.

[35] M. Grant, S. Boyd, and Y. Ye, "CVX: MATLAB software for disciplined convex programming," 2009.

[36] S. S. Keerthi and S. K. Shevade, "SMO algorithm for least-squares SVM formulations," *Neural Comput.*, vol. 15, no. 2, pp. 487–507, 2003.

[37] X. Shao, K. Wu, and B. Liao, "Single directional SMO algorithm for least squares support vector machines," *Comput. Intell. Neurosci.*, vol. 2013, 2013, Art. no. 968438.

[38] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[39] J. Alcalá-Fdez *et al.*, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *J. Multiple-Valued Log. Soft Comput.*, vol. 17, pp. 255–287, 2011.

[40] D. R. Musicant, "NDC: Normally distributed clustered datasets," 1998. [Online.] Available: www.cs.wisc.edu/dmi/svm/ndc/

[41] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

[42] C. Lian, M. Liu, J. Zhang, and D. Shen, "Hierarchical fully convolutional network for joint atrophy localization and Alzheimer's disease diagnosis using structural MRI," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 880–893, 2018.

[43] R. Y. Lo and W. J. Jagust, "Predicting missing biomarker data in a longitudinal study of Alzheimer disease," *Neurology*, vol. 78, no. 18, pp. 1376–1382, 2012.

[44] E. Westman, J.-S. Muehlboeck, and A. Simmons, "Combining MRI and CSF measures for classification of Alzheimer's disease and prediction of mild cognitive impairment conversion," *Neuroimage*, vol. 62, no. 1, pp. 229–238, 2012.

[45] M. Reuter, N. J. Schmansky, H. D. Rosas, and B. Fischl, "Within-subject template estimation for unbiased longitudinal image analysis," *Neuroimage*, vol. 61, no. 4, pp. 1402–1418, 2012.

[46] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte, "A dataset for breast cancer histopathological image classification," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 7, pp. 1455–1462, Jul. 2016.

[47] C. Gautam *et al.*, "Minimum variance-embedded deep kernel regularized least squares method for one-class classification and its applications to biomedical data," *Neural Netw.*, vol. 123, pp. 191–216, 2020.