Version of Record: https://www.sciencedirect.com/science/article/pii/S0950705122002581
Manuscript_6893c2663c3d157893252c35d2c1d353

KNN weighted reduced universum twin SVM for class imbalance learning

M.A. Ganaie[a], M. Tanveer[a,*], for the Alzheimer's Disease Neuroimaging Initiative [1]

[a] *Department of Mathematics, Indian Institute of Technology Indore, Simrol, Indore, 453552, India*

Abstract

In real world problems, imbalance of data samples poses major challenge for the classification problems as the data samples of a particular class are dominating. Problems like fault and disease detection involve imbalance data and hence need attention to avoid the bias towards a particular class. The classification models like support vector machines (SVM) get biased to majority class samples and hence results in misclassification of the minority class samples. SVM suffers as no prior information related to the data is involved in the generation of hyperplanes. Also, local information of the neighbourhood is ignored in SVM samples and thus treats each sample equally for generating the hyperplanes. However, the data points may be contaminated and may mislead the generation of hyperplanes. Inspired by the idea of prior data information and local neighbourhood information, we propose K-nearest neighbour based weighted reduced universum twin SVM for class imbalance learning (KWRUTSVM-CIL). The proposed KWRUTSVM-CIL embodies the local neighbourhood information and uses universum data to balance the classes in class imbalance problems. Local neighbourhood infor-

*Corresponding author

Email addresses: phd1901141006@iiti.ac.in (M.A. Ganaie), mtanveer@iiti.ac.in (M. Tanveer)

[1] Data used in preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf

mation is incorporated via weight matrix in the objective function. In proposed KWRUTSVM-CIL model, weight vectors are used in the corresponding constraints of the objective functions to exploit the interclass information. The oversampling and undersampling approaches are followed to balance the data in class imbalance problems. Universum data gives proir information of the data. Twin SVM, universum twin SVM, and reduced universum twin SVM for class imbalance implement empirical risk minimization principle and thus may lead to overfitting. However, the proposed KWRUTSVM-CIL model embodies regularisation term to maximise the margin and implement the structural risk minimization principle which is the marrow of statistical learning and overcomes the issues of overfitting. Experimental results and the statistical analysis signify that the generalization ability of proposed KWRUTSVM-CIL model is superior in comparison to other twin SVM based models. As an application, we use the proposed KWRUTSVM-CIL model for the diagnosis of Alzheimer's disease and breast cancer disease. The proposed KWRUTSVM-CIL model showed better generalization performance compared to other twin SVM based models in biomedical datasets.

Keywords: Universum, Rectangular kernel, Class imbalance, Imbalance ratio, Twin support vector machine, KNN weighted.

1. Introduction

Support vector machine (SVM) [1] is the successful algorithm for the classification problems used across the domains like detection of faults [2], disease detection [3, 4], face recognition [5, 6] and internet traffic classification [7]. Moreover, SVMs have also been employed in fuel consumption estimation [8] and wire arc additive manufacturing [9]. SVM via structural risk minimization principle leads to better generalization performance. However, the computational cost of SVM is higher and hence twin support vector machine (TSVM) [10] was proposed. TSVM generates two non-parallel hyperplanes by solving two smaller size quadratic programming problems (QPPs). TSVM generates each hyperplane proximal to the samples of one class and as far as possible from the samples of other class. Least squares twin SVM (LSTSVM) [11] reduced the computation cost of TSVM by solving a pair of linear system of equations instead of QPPs. Peng [12] introduced ν-twin SVM to improve the sparsity of the TSVM model while as Wang et al. [13], Yan et al. [14] introduced L_1-norm based TSVM to improve the robustness of the TSVM

model. To overcome the effects of outliers and noise, robust $L_{2,1}$-norm enhanced multi-weight vector projection SVM [15] and weighted structural twin SVM by local and global information [16] have been proposed. To further improve the generalization performance of the SVM based models, multi-view learning is exploited [17, 18]. For multiclass problems, weighted linear loss multiple birth SVM [19] has been proposed.

Weston et al. [20] introduced the concept of universum in SVM, known as USVM, which incorporates prior data distribution information for generating the classifiers. However, USVM is computationally expensive, hence, Qi et al. [21] formulated universum twin SVM (UTSVM) which is an efficient model as it solves QPPs of smaller size compared to USVM. Both USVM and UTSVM require external toolbox to solve the QPPs, hence, Xu et al. [22], Richhariya and Tanveer [23] proposed least squares twin SVM with universum data (ULSTSVM) which solves a system of linear equations for generating the classifiers. Sinz et al. [24] analysed the effect of universum and concluded that selection of universum data is problem specific. Mostly, random averaging approach is followed for the generation of universum data [21, 24]. USVM and UTSVM models incorporate universum data to improve the generalization performance of the models, however, it simultaneously increases the computation cost [3].

In addition to noise and outliers, the classification models also suffer in class imbalance problems. Here, the samples of one class (majority class) outnumber the samples of the other class (minority class). The imbalance ratio is the ratio of number of samples of majority class to the samples of minority class. For classification problems with high imbalance ratio of the samples, SVM classification models are biased to the majority class samples and hence, the samples of minority class are misclassified. In problems like diagnosis of diseases [3, 25] the focus is on the classification of minority class samples. To deal with such problem, multiple approaches have been followed. For creating the balance of classes, fuzzy SVMs for class imbalance learning [26, 27] assigned fuzzy membership weights, boosting SVM (BSVM) [28] used boosting algorithm, and entropy based fuzzy SVM (EFSVM) [29] used information entropy of data. Also, synthetic minority oversampling technique (SMOTE) [30], random under-sampling approach [31, 32], have been proposed to balance the classes. SVMs use kernel trick to handle the non-linear data. However, the aforementioned models generate the weights or synthesise the samples in the feature space instead of kernel space. To get the weights in kernel space, weighted kernel based SMOTE SVM (WKS-SVM)

[33] generated the weights in kernel space which improved the generalization performance. Minimum variance embedded weighted kernel extreme learning machine (MVWKELM) and minimum variance-embedded class-specific kernelized ELM (MVCSKELM) [34] exploits the variance of the data for the class imbalance problems. SVM has also been designed for highly imbalanced classification problems [35]. Robust fuzzy least squares TSVM for class imbalance learning (RFLSTSVM-CIL) used imbalance ratio of the samples in generating the fuzzy weights which resulted in improved generalization performance. Tanveer et al. [36], Ganaie and Tanveer [37] proposed general TSVM (Pin-GTSVM) with pinball loss function to handle the noisy data and large scale pinball loss TSVM [38] for large scale datasets. Margin maximisation in twin spheres SVM (MMTSSVM) for class imbalance data [39] generated twin hyperspheres instead of hyperplanes. To minimise the effect of noise, margin maximisation and volume minimisation in hyper-spheres machine with pinball loss [39] and K-nearest neighbor (KNN)-based margin maximisation and volume minimisation based hyper-sphere machine [40] for class imbalance problems have been proposed. Twin SVM models have also been successfully used in regression problems [41, 42, 43].

Reduced support vector machine [44] reduced the kernel matrix by randomly choosing the subset of training data to limit the number of support vectors. For regression problems, reduced twin support vector regression [45] has been proposed. Reduced kernels retain the maximum information [46] and hence, give competitive performance.

To enhance the generalization ability of a classifier and make it robust, similarity or local neighbourhood information is exploited. Half of the information about the data lies in the local neighbourhood and is exploited using KNN [47]. Weighted TSVM uses local similarity information [48] for better generalization. KNN based weighted rough ν-twin support vector machine [49], KNN based weighted TSVM [50, 51], KNN based structural TSVM [52] and KNN weighted multiclass least squares TSVM [53] exploited the local neighbourhood information by assigning different weights to the samples of each class via KNN method. Ensemble learning [54] approach has been used in TSVM based models [55, 56, 57]. For detailed review of TSVM, readers are referred to [58].

The performance of the universum based learning algorithms is better due to the prior information [20, 24]. However, incorporation of the universum data leads to increase in computational time. Recently, reduced universum TSVM for class imbalance learning (RUTSVM-CIL) [59] used universum

data efficiently to balance the samples of the different class. Here, universum data is incorporated in a manner that it give proir information about the data without increasing the computational cost of the model. Moreover, reduced kernel is used to further reduce the complexity of the models. TSVM, UTSVM and RUTSVM-CIL ignores the local neighbourhood information which leads to the problem that each sample is effecting the separating hyperplane equally, however, each sample effects the separating hyperplane differently. TSVM is not able to use the prior information, however, UTSVM can incorporate the universum data which improve the generalization performance of the model at the cost of complexity. TSVM and UTSVM models may suffer in class imbalance problems due to over representation of the majority class and under representation of the minority class. Also, TSVM, UTSVM and RUTSVM-CIL implements the empirical risk minimization principle which results in the issues of overfitting. Moreover, matrices appearing in the dual formulation of TSVM, UTSVM and RUTSVM-CIL are positive semidefinite. Hence, to overcome the aforementioned issues of TSVM, UTSVM and RUTSVM-CIL models, we formulate KNN-based weighted reduced universum TSVM for class imbalance learning (KWRUTSVM-CIL).

The major highlights of this work are

- To incorporate the local neighbourhood information, K nearest neighbour-based weights are used in the proposed KWRUTSVM-CIL.
- Unlike RUTSVM-CIL, UTSVM, TSVM and RFLSTSVM-CIL models which implement empirical risk minimization principle, the proposed KWRUTSVM-CIL model implements the structural risk minimization principle, hence, avoid the issues of overfitting.
- Similar to RUTSVM-CIL, the proposed KWRUTSVM-CIL model incorporates prior information about the data (universum data) to steer the class imbalance problem.
- Wolfe dual of the proposed KWRUTSVM-CIL involves positive definite matrices, while as the matrices in the Wolfe dual of RUTSVM-CIL, UTSVM, TSVM and RFLSTSVM-CIL are positive semidefinite.
- As an application, we use the proposed KWRUTSVM-CIL model for the diagnosis of Alzheimer's disease and breast cancer disease.

The rest of this paper is organised as follows: Section 2 discuss the existing work, Section 3 gives the formulation of the proposed model, its computational complexity and its advantages over existing models. Section 4 discusses the experimental results. Finally, the conclusions and potential future research directions are given in Section 6.

2. Related work

Suppose the data samples of minority class $+1$ are denoted by matrix $A \in R^{m_1 \times n}$ and the class -1's data points are the samples of majority class given by matrix $B \in R^{m_2 \times n}$. Also, universum data samples are given by U. Here, m_1 and m_2 are the number of samples in positive and negative class, repectively and n represents the dimensions of each sample.

2.1. Class imbalance learning

Class imbalance problems pose a real challenge to the machine learning algorithms [60]. Here, the number of samples of one class outnumber the samples of the other class, hence the classification models get biased towards the samples of majority class. Class imbalance problems are predominant in medical domain [61], software defect detection [62], computer vision [63], oil spill detection [64] and breast cancer classification [65]. The conventional classification algorithms get biased towards the dominant class of samples and the samples of other class, due to less representation, get misclassified. Particularly in medical diagnosis such as Alzheimer's disease diagnosis [66] the samples of minority class get misclassified. Hence, the tailored methods for class imbalance learning are drawing attention from the machine learning [67, 68].

Multiple techniques have been developed for the class imbalance learning [60, 69, 70]. Class imbalance learning models are broadly classified as the data level methods, algorithmic level methods and the hybrid approaches [60].

2.1.1. Data level methods

To minimize the imbalance among the samples of different classes, data level methods such as undersampling and oversampling [60, 31] have been utilized. In undersampling methods, fraction of samples from the majority class are dropped to get the balance across the samples of different classes. However, these methods such as EasyEnsemble and BalanceCascade [31] leads to

loss of information. Oversampling method replicates the samples of minority class, however, these models pose the risk of overfitting [71]. To overcome this risk, synthetic minority over-sampling technique (SMOTE) [30] generates the samples of the minority class.

2.1.2. Algorithm level methods

Algorithm level methods [35] adapt the structure of the algorithm directly to handle the class imbalance problems. Another category of algorithm level methods is the cost sensitive methods [72] which give more penalty for the misclassification of the minority class such as weighted SVM [73], weighted Lagrangian twin SVM [74], fuzzy weighted twin SVM [75] and enhanced twin SVM [76].

2.1.3. Hybrid approaches

Hybrid approaches like EasyEnsemble and BalanceCascade [31] combine the benefits of the data level methods and the algorithm level methods for handling the imbalance problems. In EasyEnsemble, random undersampling approach is followed to generate multiple balanced data subsets from the majority class, which are used to train the learners and then combine their outputs. The BalancedCascade learns the models sequentially wherein the samples of majority class being correctly classified are removed from further consideration.

2.2. K nearest neighbour (KNN) weight generation

KNN method [47] gives the local information of positive and negative classes. For each data point of +1 class, define

$$NH_s(x_k) = \{x_k^i, \text{ if both } x_k^i \text{ and } x_k \text{ are in same class }, 0 \leq i \leq m_1\}, \quad (1)$$

$$NH_d(x_k) = \{x_k^i, \text{ if } x_k^i \text{ and } x_k \text{ are in different class}, 0 \leq i \leq m_1\}, \quad (2)$$

where $NH_s(x_k)$ and $NH_d(x_k)$ represent the m_1 and m_2 neighbours of x_k from $+1$ and -1 classes, respectively. Define the adjacency matrix of class $+1$ [51], as follows

$$M_{s,ij} = \begin{cases} 1, \text{ if } x_j \in NH_s(x_i) \text{ or } x_i \in NH_s(x_j) \\ 0, \quad otherwise \end{cases} \quad (3)$$

and

$$M_{d,ij} = \begin{cases} 1, \text{ if } x_j \in NH_d(x_i) \text{ or } x_i \in NH_d(x_j) \\ 0, \quad otherwise. \end{cases} \tag{4}$$

When $M_{s,ij} = 1$ or $M_{d,ij} = 1$, undirected edge between the two samples is added in the corresponding graph. For reducing the sample points, $M_{d,ij}$ is redefined as

$$f_j = \begin{cases} 1 & \text{if } \exists i, M_{d,ij} \neq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

2.3. TSVM

TSVM [10] generates two nonparallel hyperplanes for each class of the binary classification problem. Each plane is proximal to one class of data points and as far as possible from the data points of other class.

The separating hyperplanes of TSVM for non-linear case are:

$$f_1(x) = K(x^t, C^t)u_1 + b_1 \quad \text{and} \quad f_2(x) = K(x^t, C^t)u_2 + b_2, \tag{6}$$

where $C = [A; B]$ and $K(\cdot)$ is the kernel function which is usually taken as the Gaussian kernel.

The optimization problems of TSVM are:

$$\begin{aligned} \underset{u_1,b_1,\xi_1}{min} \quad & \frac{1}{2}\|K(A, C^t)u_1 + e_1 b_1\|^2 + c_1 e_2^t \xi_1 \\ s.t. \quad & -(K(B, C^t)u_1 + e_2 b_1) + \xi_1 \geq e_2, \quad \xi_1 \geq 0 \end{aligned} \tag{7}$$

and

$$\begin{aligned} \underset{u_2,b_2,\xi_2}{min} \quad & \frac{1}{2}\|K(B, C^t)u_2 + e_2 b_2\|^2 + c_2 e_1^t \xi_2 \\ s.t. \quad & K(A, C^t)u_2 + e_1 b_2 + \xi_2 \geq e_1, \quad \xi_2 \geq 0, \end{aligned} \tag{8}$$

where c_i are penalty parameters, e_i denote vectors of ones with appropriate dimensions and ξ_i are slack vectors, for $i = 1, 2$.

Using the necessary and sufficient Karush–Kuhn–Tucker (K.K.T.) conditions, the Wolfe dual of (7) and (8) are given as follows:

$$\begin{aligned} \underset{\alpha_1}{max} \quad & e_2^t \alpha_1 - \frac{1}{2}\alpha_1^t G(H^t H)^{-1} G^t \alpha_1 \\ s.t. \quad & 0 \leq \alpha_1 \leq c_1 \end{aligned} \tag{9}$$

and

$$\begin{aligned} \max_{\alpha_2} \quad & e_1^t\alpha_2 - \frac{1}{2}\alpha_2^t H(G^tG)^{-1}H^t\alpha_2 \\ s.t. \quad & 0 \leq \alpha_2 \leq c_2, \end{aligned} \tag{10}$$

where α_1 and α_2 are the Lagrangian multipliers, $H = \begin{bmatrix} K(A, C^t) & e_1 \end{bmatrix}$ and $G = \begin{bmatrix} K(B, C^t) & e_2 \end{bmatrix}$.

Once we solve (9) and (10), the separating hyperplanes are given by

$$\begin{bmatrix} u_1 \\ b_1 \end{bmatrix} = -(H^tH)^{-1}G^t\alpha_1, \tag{11}$$

$$\begin{bmatrix} u_2 \\ b_2 \end{bmatrix} = (G^tG)^{-1}H^t\alpha_2. \tag{12}$$

As the matrices G^tG or H^tH may be ill conditioned, hence the matrices $(G^TG)^{-1}$and $(H^TH)^{-1}$ are added small positive number δ along the diagonal i.e. $(G^TG + \delta I)^{-1}$ and $(H^TH + \delta I)^{-1}$, here I is an identity matrix of appropriate dimensions.

2.4. UTSVM

UTSVM [21] incorporated the universum data into TSVM formulation. Similar to TSVM, UTSVM solves two smaller sized QPPs to generate the hyperplanes. The optimization problems of nonlinear UTSVM are given as follows

$$\begin{aligned} \min_{u_1,b_1,\xi_1,\eta_1} \quad & \frac{1}{2}\|K(A, C^t)u_1 + e_1b_1\|^2 + c_1e_2^t\xi_1 + c_ue_u^t\eta_1 \\ s.t. \quad & -(K(B, C^t)u_1 + e_2b_1) + \xi_1 \geq e_2, \quad \xi_1 \geq 0 \\ & (K(U, C^t)u_1 + e_ub_1) + \eta_1 \geq (-1+\varepsilon)e_u, \quad \eta_1 \geq 0 \end{aligned} \tag{13}$$

and

$$\begin{aligned} \min_{u_2,b_2,\xi_2,\eta_2} \quad & \frac{1}{2}\|K(B, C^t)u_2 + e_2b_2\|^2 + c_2e_1^t\xi_2 + c_ue_u^t\eta_2 \\ s.t. \quad & (K(A, C^t)u_2 + e_1b_2) + \xi_2 \geq e_1, \quad \xi_2 \geq 0 \\ & -(K(U, C^t)u_2 + e_ub_2) + \eta_2 \geq (-1+\varepsilon)e_u, \quad \eta_2 \geq 0, \end{aligned} \tag{14}$$

where c_i, c_u are penalty parameters, e_i, e_u are vectors of ones with appropriate dimensions, ξ_i, η_i are slack variables, for $i = 1, 2$ and $C = [A; B]$.

Using the necessary and sufficient K.K.T. conditions, the Wolfe dual of (13) and (14) are given as follows:

$$\begin{aligned} \underset{\alpha_1,\beta_1}{max} \;\; & e_2^t\alpha_1 - \frac{1}{2}(\alpha_1^t G - \beta_1^t O)(H^t H)^{-1}(G^t\alpha_1 - O^t\beta_1) + (\epsilon-1)e_u^t\beta_1 \\ s.t. \;\; & 0 \le \alpha_1 \le c_1, \\ & 0 \le \beta_1 \le c_u \end{aligned} \tag{15}$$

and

$$\begin{aligned} \underset{\alpha_2,\beta_2}{max} \;\; & e_2^t\alpha_2 - \frac{1}{2}(\alpha_2^t H - \beta_2^t O)(G^t G)^{-1}(H^t\alpha_2 - O^t\beta_2) + (\epsilon-1)e_u^t\beta_2 \\ s.t. \;\; & 0 \le \alpha_2 \le c_2, \\ & 0 \le \beta_2 \le c_u, \end{aligned} \tag{16}$$

where α_i and β_i are Lagrangian multipliers, for $i = 1, 2$, $H = \begin{bmatrix} K(A, C^t) & e_1 \end{bmatrix}$, $G = \begin{bmatrix} K(B, C^t) & e_2 \end{bmatrix}$, and $O = \begin{bmatrix} K(U, C^t) & e_u \end{bmatrix}$.

Once we solve (15) and (16), the separating hyperplanes are given by

$$\begin{bmatrix} u_1 \\ b_1 \end{bmatrix} = -(H^t H)^{-1}(G^t\alpha_1 - O^t\beta_1), \tag{17}$$

$$\begin{bmatrix} u_2 \\ b_2 \end{bmatrix} = (G^t G)^{-1}(H^t\alpha_2 - O^t\beta_2). \tag{18}$$

As the matrices G^tG or H^tH may be ill conditioned, hence the matrices $(G^TG)^{-1}$and $(H^TH)^{-1}$ are added small positive number δ along the diagonal i.e. $(G^TG+\delta I)^{-1}$ and $(H^TH+\delta I)^{-1}$, here I is appropriate dimension identity matrix.

The new data point $x \in R^n$ is given class label as

$$class(x) = \arg\min_{i=1,2} \frac{|K(x^t, C^t)u_i + b_i|}{\|u_i\|}. \tag{19}$$

2.5. RUTSVM-CIL

Suppose $B^* \in R^{r\times n}$ is a randomly chosen decreased data sample matrix of the majority class, here $r = m_1$. The universum data matrix is $U \in R^{d\times n}$, where $d = m_2 - m_1$. Also, $U^* \in R^{g\times n}$ be random subset of U, where $g = \lceil \frac{r}{2} \rceil$.

The optimization problem of nonlinear RUTSVM-CIL [70] are:

$$\begin{aligned} \underset{u_1,b_1,\xi_1,\psi_1}{min} \ & \frac{1}{2}\|K(A,C^t)u_1 + e_1b_1\|^2 + c_1e_1^t\xi_1 + c_ue_g^t\psi_1 \\ s.t. \ & -(K(B^*,C^t)u_1 + e_1b_1) + \xi_1 \geq e_1, \\ & (K(U^*,C^t)u_1 + e_gb_1) + \psi_1 \geq (-1+\varepsilon)e_g, \\ & \xi_1, \psi_1 \geq 0, \end{aligned} \tag{20}$$

and

$$\begin{aligned} \underset{u_2,b_2,\xi_2,\psi_2}{min} \ & \frac{1}{2}\|K(B,C^t)u_2 + e_2b_2\|^2 + c_2e_1^t\xi_2 + c_ue_r^t\psi_2 \\ s.t. \ & (K(A,C^t)u_2 + e_1b_2) + \xi_2 \geq e_1, \\ & (K(U,C^t)u_2 + e_rb_2) + \psi_2 \geq (1-\varepsilon)e_r, \\ & \xi_2, \psi_2 \geq 0, \end{aligned} \tag{21}$$

here e_1, e_g and e_r are vector of ones of appropriate dimensions.

Using the necessary and sufficient K.K.T. conditions, the Wolfe dual of (20) and (21) are given as follows:

$$\begin{aligned} \max_{\alpha_1,\,\mu_1} \ & e_1^t\alpha_1 - \frac{1}{2}(\alpha_1^tP^* - \mu_1^tO^*)(S^tS)^{-1}(P^{*t}\alpha_1 - O^{*t}\mu_1) + (\epsilon-1)e_g^t\mu_1 \\ \text{s.t. } & 0 \leq \alpha_1 \leq c_1, \\ & 0 \leq \mu_1 \leq c_u \end{aligned} \tag{22}$$

and

$$\begin{aligned} \max_{\alpha_2,\,\mu_2} \ & e_1^t\alpha_2 - \frac{1}{2}(\alpha_2^tS + \mu_2^tO)(P^tP)^{-1}(S^t\alpha_2 + O^t\mu_2) + (1-\epsilon)e_r^t\mu_2 \\ \text{s.t. } & 0 \leq \alpha_2 \leq c_2, \\ & 0 \leq \mu_2 \leq c_u, \end{aligned} \tag{23}$$

where $S = [K(A,C^t), e_1]$, $P^* = [K(B^*,C^t), e_1]$, $P = [K(B,C^t), e_2]$, $O^* = [K(U^*,C^t), e_g]$, $O = [K(U,C^t), e_r]$, $C = [A; B^*]$ is the reduced kernel and α_i, μ_i are the Lagrange multipliers, with $i = 1, 2$.

Once we solve (22) and (23), the optimal hyperplanes are given as:

$$\begin{bmatrix} u_1 \\ b_1 \end{bmatrix} = -(S^tS + \delta I)^{-1}(P^{*t}\alpha_1 - O^{*t}\mu_1), \tag{24}$$

$$\begin{bmatrix} u_2 \\ b_2 \end{bmatrix} = (P^tP + \delta I)^{-1}(S^t\alpha_2 + O^t\mu_2), \tag{25}$$

where δ is a small positive number.

The new data point $x \in R^n$ is given class label as

$$class(x) = \arg\min_{i=1,2} \frac{|K(x^t, C^t)u_i + b_i|}{\|u_i\|}. \tag{26}$$

3. Proposed KNN weighted reduced universum twin SVM for CIL (KWRUTSVM-CIL)

Motivated by [21, 70, 51], we propose KNN weighted reduced universum twin SVM for CIL (KWRUTSVM-CIL). The proposed KWRUTSVM-CIL considers the local neighbourhood information and uses universum data to balance the classes in class imbalance problems. Local neighbourhood information is incorporated via weight matrix. The weight matrix in the objective function of the proposed KWRUTSVM-CIL exploits the intraclass information and weight vector in the constraints of the corresponding objective function exploits the interclass information. The oversampling and undersampling approaches are followed to balance the samples in class imbalance problems. Universum data gives prior information of the data. Unlike TSVM, universum TSVM, and reduced universum TSVM for class imbalance, the proposed KWRUTSVM-CIL model incorporates extra regularisation term for margin maximisation and to embody the structural risk minimization principle which is the marrow of statistical learning. Also, the matrices appearing in the Wolfe dual of proposed KWRUTSVM-CIL formulation are positive definite.

3.1. Linear KWRUTSVM-CIL

The optimization problem of the proposed KWRUTSVM-CIL are

$$\begin{aligned} \min_{u_1,b_1,\xi_1,\psi_1} \ & \frac{c_3}{2}(\|u_1\|^2 + b_1^2) + \frac{1}{2}\|D_1(Au_1 + e_1b_1)\|^2 + c_1e_1^t\xi_1 + c_ue_g^t\psi_1 \\ s.t. \ & -F_1(B^*u_1 + e_1b_1) + \xi_1 \geq F_{v_1}, \quad \xi_1 \geq 0 \\ & H_1(U^*u_1 + e_gb_1) + \psi_1 \geq (-1+\varepsilon)H_g, \quad \psi_1 \geq 0 \end{aligned} \tag{27}$$

and

$$\begin{aligned} \min_{u_2,b_2,\xi_2,\psi_2} \ & \frac{c_4}{2}(\|u_2\|^2 + b_2^2) + \frac{1}{2}\|D_2(Bu_2 + e_2b_2)\|^2 + c_2e_1^t\xi_2 + c_ue_d^t\psi_2 \\ s.t. \ & F_2(Au_2 + e_1b_2) + \xi_2 \geq F_{v_2}, \quad \xi_2 \geq 0 \\ & H_2(Uu_2 + e_db_2) + \psi_2 \geq (1-\varepsilon)H_v, \quad \psi_2 \geq 0, \end{aligned} \tag{28}$$

where ξ_j, ψ_j are the slack variables, c_i and c_u are the penalty parameters, for $j = 1, 2$ and $i = 1, 2, 3, 4$. e_1, e_2, e_g and e_d are the vector of ones with appropriate dimensions. Also, $D_1 = diag(d_1, d_2, \ldots, d_{m_1})$ and $d_i = \sum_{i=1}^{m_1} M_{s,ij}$, $F_1 = diag(f_1, f_2, \ldots, f_{m_1})$, $H_1 = diag(h_1, h_2, \ldots, h_{m_g})$, F_{v_1} is a vector of diagonal elements of F_1, and similarly, H_g and H_v are defined. In the same manner, D_2, F_2, H_2 and their corresponding vectors F_{v_2}, H_v are defined.

Taking the Lagrangian of (27), we have

$$\begin{aligned} L =& \frac{c_3}{2}(\|u_1\|^2 + b_1^2) + \frac{1}{2}\|D_1(Au_1 + e_1 b_1)\|^2 + c_1 e_1^t \xi_1 + c_u e_g^t \psi_1 \\ & - \alpha^t(-F_1(B^* u_1 + e_1 b_1) + \xi_1 - F_{v_1}) \\ & - \beta^t(H_1(U^* u_1 + e_g b_1) + \psi_1 - (-1+\varepsilon)H_g) - \lambda^t \xi_1 - \sigma^t \psi_1, \end{aligned} \tag{29}$$

where α, β, λ and σ are the Lagrange multipliers.

Using the necessary and sufficient K.K.T. conditions for (29), we have

$$c_3 u_1 + (D_1 A)^t (D_1(Au_1 + e_1 b_1)) + B^{*t} F_1^t \alpha - U^{*t} H_1^t \beta = 0, \tag{30}$$
$$c_3 b_1 + (D_1 e_1)^t (D_1(Au_1 + e_1 b_1)) + e_1^t F_1^t \alpha - e_g^t H_1^t \beta = 0, \tag{31}$$
$$c_1 e_1 - \alpha - \lambda = 0, \tag{32}$$
$$c_u e_g - \beta - \sigma = 0, \tag{33}$$
$$\alpha^t(-F_1(B^* u_1 + e_1 b_1) + \xi_1 - F_{v_1}) = 0, \tag{34}$$
$$\beta^t(H_1(U^* u_1 + e_g b_1) + \psi_1 - (-1+\varepsilon)H_g) = 0, \tag{35}$$
$$\lambda^t \xi_1 = 0, \tag{36}$$
$$\sigma^t \psi_1 = 0. \tag{37}$$

Rewriting (30) and (31), we have

$$c_3 \begin{bmatrix} u_1 \\ b_1 \end{bmatrix} + \begin{bmatrix} (D_1 A)^t \\ (D_1 e_1)^t \end{bmatrix} \left(D_1 \begin{bmatrix} A & e_1 \end{bmatrix} \begin{bmatrix} u_1 \\ b_1 \end{bmatrix} \right) + \begin{bmatrix} (F_1 B^*)^t \\ (F_1 e_1)^t \end{bmatrix} \alpha - \begin{bmatrix} (H_1 U^*)^t \\ (H_1 e_g)^t \end{bmatrix} \beta = 0. \tag{38}$$

Let $S = D_1 \begin{bmatrix} A & e_1 \end{bmatrix}$, $T^* = F_1 \begin{bmatrix} B^* & e_1 \end{bmatrix}$ and $O^* = H_1 \begin{bmatrix} U^* & e_g \end{bmatrix}$, then we can write (38) as

$$(c_3 I + S^t S) \begin{bmatrix} u_1 \\ b_1 \end{bmatrix} + T^{*t} \alpha - O^{*t} \beta = 0, \tag{39}$$

$$\text{or} \quad \begin{bmatrix} u_1 \\ b_1 \end{bmatrix} = -(c_3 I + S^t S)^{-1}(T^{*t} \alpha - O^{*t} \beta). \tag{40}$$

Using (40) and the above K.K.T conditions, we have the Wolfe dual of (27) as

$$\begin{aligned} \underset{\alpha,\beta}{max} \;\; & F_{v_1}^t\alpha - \frac{1}{2}(\alpha^t T^* - \beta^t O^*)(c_3 I + S^t S)^{-1}(T^{*t}\alpha - O^{*t}\beta) + (\varepsilon - 1)H_g^t\beta \\ s.t. \;\; & 0 \leq \alpha \leq c_1, \\ & 0 \leq \beta \leq c_u. \end{aligned} \tag{41}$$

Following the similar procedure, the Wolfe dual of (28) is given as

$$\begin{aligned} \underset{\eta,\theta}{max} \;\; & F_{v_2}^t\eta - \frac{1}{2}(\eta^t S - \theta^t O)(c_4 I + T^t T)^{-1}(S^t\eta - O^t\theta) + (1 - \varepsilon)H_d^t\beta \\ s.t. \;\; & 0 \leq \eta \leq c_2, \\ & 0 \leq \theta \leq c_u, \end{aligned} \tag{42}$$

where $T = F_2 \begin{bmatrix} A & e_1 \end{bmatrix}$ and $O = H_2 \begin{bmatrix} U & e_d \end{bmatrix}$.

Once we solve (41) and (42), the separating hyperplanes are obtained as follows

$$\begin{bmatrix} u_1 \\ b_1 \end{bmatrix} = -(S^t S + c_3 I)^{-1}(T^{*t}\alpha - O^{*t}\beta), \tag{43}$$

$$\begin{bmatrix} u_2 \\ b_2 \end{bmatrix} = (T^t T + c_4 I)^{-1}(S^t\eta + O^t\theta). \tag{44}$$

The class label of a new data sample $x \in \mathbb{R}^n$ is assigned as follows:

$$class(x) = \arg\min_{i=1,2} \frac{|x^t u_i + b_i|}{\|u_i\|}. \tag{45}$$

3.2. Non-Linear KWRUTSVM-CIL

The optimization problem of the proposed KWRUTSVM-CIL for non-linear case are

$$\begin{aligned} \underset{u_1,b_1,\xi_1,\psi_1}{min} \;\; & \frac{c_3}{2}(\|u_1\|^2 + b_1^2) + \frac{1}{2}\|D_1(K(A, C^t)u_1 + e_1 b_1)\|^2 + c_1 e_1^t\xi_1 + c_u e_g^t\psi_1 \\ s.t. \;\; & -F_1(K(B^*, C^t)u_1 + e_1 b_1) + \xi_1 \geq F_{v_1}, \;\; \xi_1 \geq 0 \\ & H_1(K(U^*, C^t)u_1 + e_g b_1) + \psi_1 \geq (-1 + \varepsilon)H_g, \;\; \psi_1 \geq 0 \end{aligned} \tag{46}$$

and

$$\begin{aligned}\underset{u_2,b_2,\xi_2,\psi_2}{min}\ & \frac{c_4}{2}(\|u_2\|^2+b_2^2)+\frac{1}{2}\|D_2(K(B,C^t)u_2+e_2b_2)\|^2+c_2e_1^t\xi_2+c_ue_d^t\psi_2\\ s.t.\ & F_2(K(A,C^t)u_2+e_1b_2)+\xi_2\geq F_{v_2},\ \ \xi_2\geq 0\\ & H_2(K(U,C^t)u_2+e_db_2)+\psi_2\geq(1-\varepsilon)H_v,\ \ \psi_2\geq 0,\end{aligned}\tag{47}$$

where $C=[A;B^*]$ and $K(\cdot)$ is the reduced kernel.

Following the similar procedure as in the linear case, the Wolfe duals of (46) and (47) are

$$\begin{aligned}\underset{\alpha,\beta}{max}\ & F_{v_1}^t\alpha-\frac{1}{2}(\alpha^tT^*-\beta^tO^*)(c_3I+S^tS)^{-1}(T^{*t}\alpha-O^{*t}\beta)+(\varepsilon-1)H_g^t\beta\\ s.t.\ & 0\leq\alpha\leq c_1,\ 0\leq\beta\leq c_u\end{aligned}\tag{48}$$

and

$$\begin{aligned}\underset{\eta,\theta}{max}\ & F_{v_2}^t\eta-\frac{1}{2}(\eta^tS-\theta^tO)(c_4I+T^tT)^{-1}(S^t\eta-O^t\theta)+(1-\varepsilon)H_d^t\beta\\ s.t.\ & 0\leq\eta\leq c_2,\ 0\leq\theta\leq c_u,\end{aligned}\tag{49}$$

where $S=D_1\begin{bmatrix}K(A,C^t) & e_1\end{bmatrix}$, $T^*=F_1\begin{bmatrix}K(B^*,C^t) & e_1\end{bmatrix}$, $O^*=H_1\begin{bmatrix}K(U^*,C^t) & e_g\end{bmatrix}$, $T=F_2\begin{bmatrix}K(A,C^t) & e_1\end{bmatrix}$ and $O=H_2\begin{bmatrix}K(U,C^t) & e_d\end{bmatrix}$.

Once we solve (48) and (49), the separating hyperplanes are obtained as follows:

$$\begin{bmatrix}u_1\\ b_1\end{bmatrix}=-(S^tS+c_3I)^{-1}(T^{*t}\alpha-O^{*t}\beta),\tag{50}$$

$$\begin{bmatrix}u_2\\ b_2\end{bmatrix}=(T^tT+c_4I)^{-1}(S^t\eta+O^t\theta).\tag{51}$$

The class label of a new data sample $x\in\mathbb{R}^n$ is assigned as follows:

$$class(x)=\arg\min_{i=1,2}\frac{|K(x^t,C^t)u_i+b_i|}{\|u_i\|}.\tag{52}$$

3.3. Computational complexity

Consider a binary classification problem with m number of samples such that m_1 samples belong to the positive class and m_2 samples belong to the negative class ($m=m_1+m_2$). Assume that we have the class imbalance problem with imbalance ratio, $IR=\frac{m_2}{m_1}$. The time complexity of the TSVM

is $(IR^3 + 1)O(m_1)^3$ and RUTSVM-CIL is $(IR^3 + 3.375)O(m_1)^3$ [70]. The computational complexity of proposed KWRUTSVM-CIL involves the assignment of weights to the samples via KNN graph and solving the QPPs. Thus, main complexity of the model involves the following two steps:

1. Weight initialization via KNN graph: Using $O(m^2log(m))$ the weights of the weight matrix D with m points can be computed.
2. The complexity of optimizing the the QPPs of the proposed formulations is same as that of the RUTSVM-CIL model.

Hence, the complexity of the proposed model is relatively increased compared to RUTSVM-CIL model due to the incorporation of the local neighborhood information.

3.4. Analysis of proposed KWRUTSVM-CIL with respect to TSVM, UTSVM and RUTSVM-CIL models

The advantages of the proposed KWRUTSVM-CIL with respect to TSVM, UTSVM and RUTSVM-CIL models are given as follows

- TSVM, UTSVM and RUTSVM-CIL minimize empirical risk which may result in the issues of overfitting. However, the proposed KWRUTSVM-CIL model minimizes structural risk which overcomes the issues of overfitting. Also, TSVM, UTSVM and RUTSVM-CIL assume that data matrices appearing in the formulation are positive definite. However, in real world scenarios, such assumption may not be valid which results in the issues of singularity. The matrices in the proposed KWRUTSVM-CIL are positive definite and hence, theoretically more stable.

- TSVM, UTSVM and RUTSVM-CIL assume that all the samples are equally important for generating hyperplanes. However, such an assumption may not hold in real world scenarios. Thus, the proposed KWRUTSVM-CIL model exploit neighbourhood information via KNN to give the appropriate weights to the samples which improves the generalization performance.

4. Experimental Results

In classification problems, class imbalance problem poses a challenge to the classification algorithms. Such kind of problems have irregularity in the probability distribution of samples corresponding to different classes. These

kind of datasets are termed as imbalance datasets [60]. The classification algorithms suffer in problems dominated by the samples of a particular class. The ratio of number of majority class samples to the number of minority class samples is known as imbalance ratio (IR). Mathematically,

$$IR = \frac{\text{Count of majority class data points}}{\text{Count of minority class data points}}. \tag{53}$$

We analyse the classification performance of TSVM [10], UTSVM [21], RUTSVM-CIL [70], MVKWELM [34], MVCSKELM [34], KW-SMOTE-SVM [33], RFLSTSVM-CIL [77] and proposed KWRUTSVM-CIL model. All the models are evaluated on datasets from UCI machine learning repository [78] and KEEL repository [79]. We follow 5-fold cross validation and the average results of different classification models is given in Table 2. Here, the data is partitioned randomly into 5 subsets out of which one partition reserved for testing and the rest partitions involved in training the model. This process is repeated 5 times and average accuracy of the folds is taken as the performance measure. All the classification models have been implemented in MATLAB R2017b on a system with configuration Intel (R) Core (TM) i7-6700 CPU @ 3.40 GHZ with 8 GB of RAM. We use Gaussian kernel $K(x, y) = exp^{-\frac{1}{2\mu^2}\|x-y\|^2}$, where μ is the kernel parameter.

The classification models involve proper tuning of parameters for better performance. We used standard grid search approach to tune the optimal from the following range of parameters: $c_0 = [0.5, 1, 1.5, 2, 2.5]$, $\mu = [10^{-5}, 10^{-4}, \ldots, 10^4, 10^5]$, $c_u = c_i = [10^{-5}, 10^{-4}, \ldots, 10^4, 10^5]$, for $i = 1, 2, 3, 4$. To reduce the computations, we use $c_1 = c_2 = c_u, c_3 = c_4$.

For generating the universum data, we used random averaging method [70]. In random averaging method, same number of samples are randomly chosen from the given classes and the selected samples are averaged to get the universum data. For generating the KNN weights [53], we followed procedure given above in Subsection 2.2.

The evaluation of the baseline classifiers and the proposed KWRUTSVM-CIL is performed using area under receiver operating characteristics (ROC)

curve i.e. (AUC) or accuracy and other measures which are defined as:

$$\text{Accuracy, AUC} = \frac{TP + TN}{TP + FP + TN + FN}, \tag{54}$$

$$\text{Sensitivity or Recall} = \frac{TP}{TP + FN}, \tag{55}$$

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{56}$$

$$\text{F-measure} = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{57}$$

$$\text{G-mean} = \sqrt{Precision \times Recall}, \tag{58}$$

where TP denotes true positive, TN : true negative, FP : false positive and FN : false negative. The different hyperparameters of a given classifier generates multiple classifiers which are represented by a point in ROC curve. The optimal hyperparameters are the parameters corresponding to the point (a classifier in ROC curve) in northwest of the AUC curve. The optimal point (classifier) is the classifier corresponding to the optimal hyperparameters and the same information is reported in above given measures.

Alzheimer's data is obtained from Alzheimer's Disease Neuroimaging Initiative (ADNI) database `http://adni.loni.usc.edu/`. In 2003, ADNI with Michael W. Weiner, MD as the Principal investigator was established. The motive of ADNI is to analyse the onset of Alzheimer's disease via Neuroimaging approaches involving positron emission tomography (PET), magnetic resonance imaging (MRI) and other biological markers. For details, we refer to `www.adni-info.org`. We used ADNI baseline dataset [80, 81] and downloaded 817 structural MRI (sMRI) images. For Volume based morphometry (VolBM) of the images, we used Freesurfer's recon-all pipeline (version6.0.1) [82, 83]. Out of 817 images, 4 images failed to process via Freesurfer pipeline. Thus, we use control normal (CN), mild cognitive impairment (MCI), and Alzheimer's disease (AD) subjects with $228, 398$ and 187 images respectively.

The BreakHis histopathological breast cancer image dataset [84] contains 1240 images with $400X$ magnification. The dataset involves 2 classes: malignant class and benign class. Benign category is subdivided into phyllodes tumor (PT), adenosis (ADN), tubular adenoma (TA) and fibroadenoma (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 contains $169, 208, 138$ and

137 images, respectively. The images are converted into gray level images and up to 3 levels of decomposition of Daubechies-4 (db4) wavelet is applied on each image [85]. The concatenation of approximation coefficients and detail coefficients is used as the feature vector. The information of biomedical data is presented in Table 9.

4.1. Real world datasets

The experimental results of TSVM [10], UTSVM [21], RUTSVM-CIL [70], MVKWELM [34], MVCSKELM [34], KW-SMOTE-SVM [33], RFLSTSVM-CIL [77] and proposed KWRUTSVM-CIL with Gaussian kernel are presented in Table 2.

From Table 2, the average accuracy of TSVM [10], UTSVM [21], RUTSVM-CIL [70], MVKWELM [34], MVCSKELM [34], KW-SMOTE-SVM [33], RFLSTSVM-CIL [77] and the proposed KWRUTSVM-CIL classifiers are 0.8891, 0.8952, 0.893, 0.8364, 0.8854, 0.8539, 0.895 and 0.9028, respectively. The proposed KWRUTSVM-CIL classifier demonstrated superior performance in comparison with the existing models. According to overall win-tie-loss analysis, the TSVM model wins in 2 datasets datasets, UTSVM wins in 6 datasets, RUTSVM-CIL models win in 1 dataset, KW-SMOTE-SVM and RFLSTSVM-CIL win in 3 and 5 datasets, respectively. The proposed KWRUTSVM-CIL model wins in 20 datasets. Thus, the proposed KWRUTSVM-CIL model is the successful classifier with majority of the wins. Since average accuracy may be biased as higher performance in one dataset may compensate the others, hence, we rank the classification models on each dataset to evaluate them. From Table 2, one can see that the average rank of TSVM [10], UTSVM [21], RUTSVM-CIL [70], MVKWELM [34], MVCSKELM [34], KW-SMOTE-SVM [33], RFLSTSVM-CIL [77] and proposed KWRUTSVM-CIL are 4.29, 3.6, 3.84, 7.51, 4.79, 6.51, 3.49, and 1.98, respectively. Thus, the proposed KWRUTSVM-CIL model emerged as the best classifier with lowest average rank.

Table 1: Dataset details

Datasets	Samples × Features	IR
abalone9-18	731 × 8	16.4048
aus	690 × 15	1.2476
brwisconsin	683 × 10	1.8577
bupa_or_liver-disorders	345 × 7	1.3793
checkerboard_Data	690 × 15	1.2476
cleve	297 × 14	1.1679
cmc	1473 × 10	1.3418
ecoli-0-1_vs_2-3-5	244 × 8	9.1667
ecoli-0-1_vs_5	240 × 7	11
ecoli-0-1-4-6_vs_5	280 × 7	13
ecoli-0-1-4-7_vs_2-3-5-6	336 × 8	10.5862
ecoli-0-1-4-7_vs_5-6	332 × 7	12.28
ecoli-0-2-3-4_vs_5	202 × 8	9.1
ecoli-0-3-4-6_vs_5	205 × 8	9.25
ecoli-0-3-4-7_vs_5-6	257 × 8	9.28
ecoli-0-4-6_vs_5	203 × 7	9.15
ecoli-0-6-7_vs_3-5	222 × 8	9.0909
ecoli-0-6-7_vs_5	220 × 7	10
ecoli0137vs26	311 × 8	4.7593
ecoli01vs5	240 × 8	11
ecoli2	336 × 8	8.6
ecoli3	336 × 8	8.6
ecoli4	336 × 8	15.8
glass4	214 × 10	15.4615
heart-stat	270 × 14	1.25
iono	351 × 34	1.7857
new-thyroid1	215 × 6	5.1429
pima	768 × 9	1.8657
vehicle1	846 × 19	2.8986
vehicle2	846 × 19	2.8807
votes	435 × 17	1.5893
vowel	988 × 11	9.9778
wpbc	194 × 34	3.2174
yeast-0-2-5-6_vs_3-7-8-9	1004 × 9	9.1414
yeast-0-2-5-7-9_vs_3-6-8	1004 × 9	9.1414
yeast-0-5-6-7-9_vs_4	528 × 9	9.3529
yeast-2_vs_4	514 × 9	9.0784
yeast2vs8	483 × 9	23.15
yeast3	1484 × 9	8.1043
yeast5	1484 × 9	32.7273

Table 2: Classification performance of baseline models and proposed KWRUTSVM-CIL model on real world KEEL datasets with Gaussian kernel.

Datasets	TSVM [10]	UTSVM [21]	RUTSVM-CIL [70]	MVKWELM [34]	MVCSKELM [34]	KW-SMOTE-SVM [33]	RFLSTSVM-CIL [77]	KWRUTSVM-CIL
	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean
abalone9-18	0.79 ± 0.03 $0.5, 0.52$	0.82 ± 0.06 $0.46, 0.5$	0.88 ± 0.06 $0.51, 0.56$	0.66 ± 0.07 $0.18, 0.27$	0.76 ± 0.06 $0.4, 0.43$	0.87 ± 0.06 $0.39, 0.48$	0.85 ± 0.03 $0.42, 0.49$	$\mathbf{0.89} \pm 0.04$ $0.53, 0.58$
aus	0.87 ± 0.04 $0.86, 0.86$	0.87 ± 0.02 $0.86, 0.86$	0.88 ± 0.06 $0.87, 0.87$	0.87 ± 0.03 $0.86, 0.86$	0.87 ± 0.02 $0.86, 0.86$	0.88 ± 0.04 $0.87, 0.87$	0.87 ± 0.04 $0.86, 0.86$	$\mathbf{0.89} \pm 0.05$ $0.88, 0.88$
brwisconsin	$\mathbf{0.98} \pm 0.02$ $0.97, 0.97$	$\mathbf{0.98} \pm 0.02$ $0.97, 0.97$	$\mathbf{0.98} \pm 0.01$ $0.97, 0.97$	0.97 ± 0.02 $0.96, 0.96$	$\mathbf{0.98} \pm 0.01$ $0.96, 0.96$	0.97 ± 0.02 $0.96, 0.96$	$\mathbf{0.98} \pm 0.01$ $0.97, 0.97$	$\mathbf{0.98} \pm 0.01$ $0.97, 0.97$
bupa_or_liver-disorders	$\mathbf{0.71} \pm 0.05$ $0.67, 0.67$	$\mathbf{0.71} \pm 0.09$ $0.64, 0.65$	0.69 ± 0.08 $0.67, 0.68$	0.59 ± 0.04 $0.54, 0.54$	$\mathbf{0.71} \pm 0.06$ $0.66, 0.66$	0.67 ± 0.02 $0.64, 0.64$	0.7 ± 0.05 $0.65, 0.65$	$\mathbf{0.71} \pm 0.07$ $0.66, 0.66$
checkerboard	0.87 ± 0.04 $0.86, 0.86$	0.87 ± 0.04 $0.86, 0.86$	0.88 ± 0.06 $0.87, 0.87$	0.87 ± 0.03 $0.86, 0.86$	0.87 ± 0.02 $0.86, 0.86$	0.88 ± 0.04 $0.87, 0.87$	0.87 ± 0.04 $0.86, 0.86$	$\mathbf{0.89} \pm 0.05$ $0.88, 0.88$
cleve	0.83 ± 0.06 $0.82, 0.82$	0.84 ± 0.05 $0.83, 0.83$	$\mathbf{0.86} \pm 0.04$ $0.85, 0.85$	0.82 ± 0.09 $0.81, 0.81$	0.82 ± 0.04 $0.8, 0.8$	0.83 ± 0.06 $0.82, 0.82$	0.83 ± 0.08 $0.81, 0.81$	0.85 ± 0.03 $0.83, 0.84$
cmc	$\mathbf{0.7} \pm 0.03$ $0.63, 0.63$	$\mathbf{0.7} \pm 0.02$ $0.63, 0.63$	0.69 ± 0.02 $0.63, 0.64$	0.69 ± 0.04 $0.64, 0.64$	0.69 ± 0.03 $0.66, 0.66$	0.66 ± 0.01 $0.62, 0.62$	$\mathbf{0.7} \pm 0.04$ $0.64, 0.64$	$\mathbf{0.7} \pm 0.02$ $0.64, 0.65$
ecoli-0-1_vs_2-3-5	0.9 ± 0.04 $0.8, 0.81$	$\mathbf{0.93} \pm 0.06$ $0.88, 0.88$	0.9 ± 0.07 $0.63, 0.66$	0.87 ± 0.08 $0.61, 0.64$	0.89 ± 0.05 $0.71, 0.72$	0.85 ± 0.05 $0.7, 0.71$	0.92 ± 0.05 $0.82, 0.83$	0.92 ± 0.05 $0.8, 0.82$
ecoli-0-1_vs_5	0.92 ± 0.07 $0.87, 0.88$	$\mathbf{0.96} \pm 0.06$ $0.84, 0.85$	0.94 ± 0.11 $0.87, 0.88$	0.9 ± 0.12 $0.61, 0.64$	0.95 ± 0.07 $0.92, 0.92$	0.91 ± 0.07 $0.78, 0.79$	$\mathbf{0.96} \pm 0.06$ $0.86, 0.86$	$\mathbf{0.96} \pm 0.05$ $0.86, 0.87$

Continued on next page

Table 2 – continued from previous page

Datasets	TSVM [10]	UTSVM [21]	RUTSVM-CIL [70]	MVKWELM [34]	MVCSKELM [34]	KW-SMOTE-SVM [33]	RFLSTSVM-CIL [77]	KWRUTSVM-CIL
	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean
ecoli-0-1-4-6_vs_5	0.92 ± 0.07	0.92 ± 0.07	0.92 ± 0.12	0.91 ± 0.07	0.94 ± 0.08	0.87 ± 0.09	0.93 ± 0.07	$\mathbf{0.95} \pm 0.06$
	0.89, 0.9	0.89, 0.9	0.89, 0.89	0.6, 0.63	0.85, 0.85	0.83, 0.84	0.91, 0.92	0.76, 0.78
ecoli-0-1-4-7_vs_2-3-5-6	0.91 ± 0.08	0.89 ± 0.02	0.91 ± 0.06	0.86 ± 0.12	0.91 ± 0.08	0.84 ± 0.09	$\mathbf{0.92} \pm 0.07$	0.91 ± 0.05
	0.83, 0.83	0.74, 0.75	0.68, 0.7	0.6, 0.62	0.85, 0.85	0.43, 0.5	0.7, 0.72	0.77, 0.78
ecoli-0-1-4-7_vs_5-6	0.92 ± 0.05	$\mathbf{0.94} \pm 0.05$	0.93 ± 0.05	0.89 ± 0.05	0.92 ± 0.07	0.85 ± 0.03	0.92 ± 0.05	0.93 ± 0.05
	0.86, 0.86	0.75, 0.77	0.72, 0.74	0.64, 0.66	0.65, 0.68	0.4, 0.48	0.77, 0.78	0.73, 0.75
ecoli-0-2-3-4_vs_5	0.92 ± 0.11	0.95 ± 0.06	0.92 ± 0.12	0.9 ± 0.08	0.94 ± 0.11	0.91 ± 0.07	0.95 ± 0.06	$\mathbf{0.96} \pm 0.06$
	0.85, 0.85	0.79, 0.8	0.85, 0.85	0.65, 0.68	0.84, 0.85	0.81, 0.82	0.79, 0.8	0.85, 0.86
ecoli-0-3-4-6_vs_5	0.92 ± 0.12	0.94 ± 0.12	0.92 ± 0.05	0.89 ± 0.06	0.91 ± 0.11	0.89 ± 0.12	0.95 ± 0.05	$\mathbf{0.96} \pm 0.06$
	0.86, 0.87	0.89, 0.89	0.66, 0.69	0.67, 0.69	0.79, 0.8	0.8, 0.8	0.82, 0.83	0.88, 0.89
ecoli-0-3-4-7_vs_5-6	0.93 ± 0.05	$\mathbf{0.94} \pm 0.05$	0.92 ± 0.07	0.86 ± 0.08	0.92 ± 0.04	0.89 ± 0.01	0.92 ± 0.08	0.93 ± 0.08
	0.85, 0.85	0.8, 0.81	0.69, 0.72	0.64, 0.66	0.69, 0.72	0.82, 0.82	0.87, 0.88	0.75, 0.77
ecoli-0-4-6_vs_5	0.94 ± 0.11	0.95 ± 0.07	0.93 ± 0.08	0.89 ± 0.1	0.93 ± 0.05	0.91 ± 0.11	$\mathbf{0.96} \pm 0.06$	0.95 ± 0.07
	0.85, 0.86	0.92, 0.92	0.79, 0.8	0.68, 0.71	0.7, 0.74	0.82, 0.83	0.9, 0.91	0.92, 0.92
ecoli-0-6-7_vs_3-5	0.9 ± 0.16	0.91 ± 0.11	0.91 ± 0.05	0.87 ± 0.09	0.89 ± 0.09	0.81 ± 0.1	0.92 ± 0.12	$\mathbf{0.93} \pm 0.1$
	0.81, 0.84	0.81, 0.82	0.76, 0.78	0.67, 0.69	0.67, 0.7	0.45, 0.51	0.72, 0.73	0.78, 0.81

Continued on next page

Table 2 – continued from previous page

Datasets	TSVM [10]	UTSVM [21]	RUTSVM-CIL [70]	MVKWELM [34]	MVCSKELM [34]	KW-SMOTE-SVM [33]	RFLSTSVM-CIL [77]	KWRUTSVM-CIL
	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean
ecoli-0-6-7_vs_5	0.91 ± 0.11	0.9 ± 0.07	$\mathbf{0.92} \pm 0.07$	0.86 ± 0.1	0.88 ± 0.06	0.85 ± 0.11	0.89 ± 0.11	$\mathbf{0.92} \pm 0.07$
	0.79, 0.8	0.75, 0.76	0.71, 0.74	0.63, 0.65	0.58, 0.63	0.77, 0.78	0.82, 0.82	0.75, 0.77
ecoli0137vs26	0.93 ± 0.03	0.93 ± 0.06	$\mathbf{0.94} \pm 0.03$	0.89 ± 0.03	$\mathbf{0.94} \pm 0.02$	0.88 ± 0.04	$\mathbf{0.94} \pm 0.02$	$\mathbf{0.94} \pm 0.03$
	0.9, 0.9	0.89, 0.89	0.88, 0.88	0.74, 0.75	0.91, 0.91	0.68, 0.71	0.86, 0.86	0.89, 0.89
ecoli01vs5	$\mathbf{1} \pm 0.01$	$\mathbf{1} \pm 0$	$\mathbf{1} \pm 0.01$	0.98 ± 0.02	0.99 ± 0.01	0.99 ± 0.01	$\mathbf{1} \pm 0.01$	$\mathbf{1} \pm 0.01$
	0.96, 0.96	1, 1	0.98, 0.98	0.82, 0.84	0.92, 0.93	0.94, 0.94	0.98, 0.98	0.98, 0.98
ecoli2	0.9 ± 0.06	0.89 ± 0.1	0.9 ± 0.03	0.87 ± 0.05	0.91 ± 0.06	0.83 ± 0.04	0.91 ± 0.05	$\mathbf{0.92} \pm 0.03$
	0.66, 0.68	0.66, 0.69	0.6, 0.65	0.49, 0.56	0.62, 0.66	0.44, 0.52	0.7, 0.72	0.67, 0.7
ecoli3	0.9 ± 0.06	0.91 ± 0.04	0.9 ± 0.03	0.87 ± 0.05	0.91 ± 0.06	0.83 ± 0.04	0.91 ± 0.05	$\mathbf{0.92} \pm 0.03$
	0.66, 0.68	0.7, 0.72	0.6, 0.65	0.49, 0.56	0.62, 0.66	0.44, 0.52	0.7, 0.72	0.67, 0.7
ecoli4	0.97 ± 0.02	$\mathbf{0.98} \pm 0$	$\mathbf{0.98} \pm 0.01$	0.92 ± 0.01	0.95 ± 0.02	0.91 ± 0.03	0.95 ± 0.06	$\mathbf{0.98} \pm 0.01$
	0.68, 0.72	0.77, 0.79	0.78, 0.8	0.43, 0.52	0.59, 0.65	0.43, 0.52	0.69, 0.72	0.79, 0.81
glass4	$\mathbf{0.96} \pm 0.07$	0.94 ± 0.12	$\mathbf{0.96} \pm 0.07$	0.91 ± 0.08	0.95 ± 0.07	0.81 ± 0.11	$\mathbf{0.96} \pm 0.08$	$\mathbf{0.96} \pm 0.04$
	0.91, 0.92	0.85, 0.85	0.83, 0.84	0.53, 0.59	0.75, 0.77	0.39, 0.45	0.88, 0.89	0.7, 0.74
heart-stat	0.82 ± 0.04	0.82 ± 0.06	0.78 ± 0.03	0.59 ± 0.03	0.8 ± 0.03	$\mathbf{0.84} \pm 0.04$	0.8 ± 0.04	0.82 ± 0.04
	0.8, 0.8	0.79, 0.79	0.76, 0.76	0.55, 0.56	0.77, 0.78	0.82, 0.82	0.77, 0.77	0.79, 0.8
iono	$\mathbf{0.95} \pm 0.04$	$\mathbf{0.95} \pm 0.03$	0.91 ± 0.03	0.84 ± 0.03	0.93 ± 0.04	0.88 ± 0.03	$\mathbf{0.95} \pm 0.02$	0.94 ± 0.02
	0.94, 0.94	0.93, 0.93	0.89, 0.9	0.8, 0.81	0.92, 0.92	0.86, 0.87	0.94, 0.94	0.91, 0.91
new-thyroid1	$\mathbf{1} \pm 0.01$	0.99 ± 0.01	$\mathbf{1} \pm 0$	0.97 ± 0.02	0.99 ± 0.01	0.9 ± 0.05	$\mathbf{1} \pm 0.01$	$\mathbf{1} \pm 0$
	0.99, 0.99	0.95, 0.95	1, 1	0.9, 0.9	0.97, 0.97	0.75, 0.77	0.99, 0.99	1, 1
pima	0.72 ± 0.07	0.74 ± 0.04	0.74 ± 0.05	0.66 ± 0.04	0.72 ± 0.07	$\mathbf{0.75} \pm 0.02$	0.73 ± 0.04	0.74 ± 0.04
	0.62, 0.62	0.66, 0.68	0.67, 0.68	0.57, 0.57	0.63, 0.64	0.67, 0.67	0.66, 0.67	0.67, 0.68

Continued on next page

Table 2 – continued from previous page

Datasets	TSVM [10]	UTSVM [21]	RUTSVM-CIL [70]	MVKWELM [34]	MVCSKELM [34]	KW-SMOTE-SVM [33]	RFLSTSVM-CIL [77]	KWRUTSVM-CIL
	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean
vehicle1	0.85 ± 0.02 $0.72, 0.74$	0.84 ± 0.02 $0.7, 0.72$	0.85 ± 0.02 $0.71, 0.73$	0.61 ± 0.04 $0.44, 0.44$	0.82 ± 0.04 $0.67, 0.7$	0.78 ± 0.02 $0.64, 0.66$	0.82 ± 0.03 $0.69, 0.71$	$\mathbf{0.86} \pm 0.02$ $0.73, 0.75$
vehicle2	$\mathbf{0.99} \pm 0$ $0.98, 0.98$	0.98 ± 0.02 $0.97, 0.97$	0.98 ± 0.01 $0.97, 0.97$	0.9 ± 0.03 $0.82, 0.82$	0.97 ± 0.02 $0.95, 0.95$	0.93 ± 0.01 $0.83, 0.84$	0.98 ± 0.01 $0.97, 0.97$	$\mathbf{0.99} \pm 0.02$ $0.97, 0.97$
votes	$\mathbf{0.97} \pm 0.02$ $0.95, 0.95$	$\mathbf{0.97} \pm 0.02$ $0.96, 0.96$	$\mathbf{0.97} \pm 0.01$ $0.96, 0.96$	0.94 ± 0.02 $0.93, 0.93$	0.96 ± 0.02 $0.95, 0.95$	0.96 ± 0.01 $0.95, 0.95$	0.96 ± 0.02 $0.95, 0.95$	$\mathbf{0.97} \pm 0.02$ $0.96, 0.96$
vowel	$\mathbf{1} \pm 0$ $1, 1$	$\mathbf{1} \pm 0$ $1, 1$	0.99 ± 0.01 $0.99, 0.99$	0.94 ± 0.01 $0.62, 0.67$	$\mathbf{1} \pm 0$ $1, 1$	0.89 ± 0.04 $0.81, 0.81$	$\mathbf{1} \pm 0$ $1, 1$	$\mathbf{1} \pm 0$ $0.98, 0.98$
wpbc	0.68 ± 0.09 $0.51, 0.51$	$\mathbf{0.72} \pm 0.06$ $0.55, 0.57$	0.69 ± 0.05 $0.51, 0.54$	0.58 ± 0.12 $0.39, 0.41$	0.7 ± 0.1 $0.52, 0.55$	$\mathbf{0.72} \pm 0.08$ $0.54, 0.57$	0.7 ± 0.09 $0.53, 0.54$	0.69 ± 0.07 $0.52, 0.54$
yeast-0-2-5-6_vs_3-7-8-9	0.78 ± 0.04 $0.64, 0.64$	0.79 ± 0.04 $0.58, 0.59$	0.81 ± 0.03 $0.6, 0.6$	0.78 ± 0.04 $0.43, 0.47$	0.81 ± 0.05 $0.59, 0.59$	0.72 ± 0.11 $0.38, 0.43$	0.81 ± 0.05 $0.59, 0.59$	$\mathbf{0.82} \pm 0.02$ $0.63, 0.63$
yeast-0-2-5-7-9_vs_3-6-8	$\mathbf{0.92} \pm 0.03$ $0.77, 0.78$	0.91 ± 0.03 $0.8, 0.8$	0.91 ± 0.04 $0.77, 0.78$	0.85 ± 0.05 $0.52, 0.56$	0.91 ± 0.02 $0.75, 0.76$	0.89 ± 0.04 $0.62, 0.65$	$\mathbf{0.92} \pm 0.03$ $0.73, 0.74$	$\mathbf{0.92} \pm 0.04$ $0.79, 0.79$
yeast-0-5-6-7-9_vs_4	0.81 ± 0.08 $0.55, 0.56$	0.8 ± 0.06 $0.43, 0.49$	$\mathbf{0.83} \pm 0.1$ $0.53, 0.56$	0.8 ± 0.04 $0.42, 0.48$	0.82 ± 0.07 $0.46, 0.51$	0.76 ± 0.04 $0.41, 0.45$	0.81 ± 0.05 $0.52, 0.55$	$\mathbf{0.83} \pm 0.11$ $0.56, 0.59$
yeast-2_vs_4	$\mathbf{0.91} \pm 0.07$ $0.69, 0.71$	$\mathbf{0.91} \pm 0.05$ $0.76, 0.77$	0.89 ± 0.05 $0.71, 0.72$	0.85 ± 0.06 $0.53, 0.57$	0.89 ± 0.07 $0.76, 0.76$	0.89 ± 0.06 $0.77, 0.77$	$\mathbf{0.91} \pm 0.06$ $0.74, 0.75$	0.89 ± 0.04 $0.78, 0.79$
yeast2vs8	$\mathbf{0.82} \pm 0.14$ $0.64, 0.65$	$\mathbf{0.82} \pm 0.14$ $0.61, 0.62$	0.81 ± 0.14 $0.57, 0.58$	0.77 ± 0.11 $0.68, 0.7$	0.79 ± 0.07 $0.41, 0.44$	0.77 ± 0.11 $0.68, 0.7$	$\mathbf{0.82} \pm 0.14$ $0.72, 0.74$	0.8 ± 0.14 $0.7, 0.73$
yeast3	0.92 ± 0.04	0.92 ± 0.04	$\mathbf{0.93} \pm 0.03$	0.85 ± 0.03	0.91 ± 0.03	0.89 ± 0.04	0.92 ± 0.02	$\mathbf{0.93} \pm 0.02$

Continued on next page

Table 2 – continued from previous page

Datasets	TSVM [10]	UTSVM [21]	RUTSVM-CIL [70]	MVKWELM [34]	MVCSKELM [34]	KW-SMOTE-SVM [33]	RFLSTSVM-CIL [77]	KWRUTSVM-CIL
	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean
yeast5	0.72, 0.74 0.97 ± 0.01 0.49, 0.57	0.75, 0.77 0.97 ± 0.03 0.61, 0.66	0.72, 0.74 0.97 ± 0.01 0.54, 0.61	0.5, 0.56 0.92 ± 0.01 0.29, 0.41	0.68, 0.7 0.97 ± 0.01 0.55, 0.62	0.75, 0.75 **0.98** ± 0.01 0.6, 0.66	0.7, 0.72 **0.98** ± 0.01 0.67, 0.71	0.74, 0.76 **0.98** ± 0.01 0.61, 0.66
Average AUC	0.8891	0.8952	0.893	0.8364	0.8854	0.8539	0.895	**0.9028**
Average Rank	4.29	3.6	3.84	7.51	4.79	6.51	3.49	**1.98**
Overall Win-Tie-Loss	[2, 0, 0]	[6, 0, 0]	[1, 0, 0]	[0, 0, 20]	[0, 0, 0]	[3, 0, 17]	[5, 0, 2]	**[20,0,0]**

Here, STD denotes the standard deviation.

Table 3: Nemenyi post hoc significant difference of classification models based on their performance in real world KEEL datasets with Gaussian kernel.

Significance	TSVM [10]	UTSVM [21]	RUTSVM-CIL [70]	MVKWELM [34]	MVCSKELM [34]	KW-SMOTE-SVM [33]	RFLSTSVM-CIL [77]
Proposed	Yes	No	Yes	Yes	Yes	Yes	No

Table 4: Nemenyi post hoc significant difference of classification models based on their performance in biomedical datasets with Gaussian kernel

Significance	TSVM [10]	UTSVM [21]	RUTSVM-CIL [70]	MVKWELM [34]	MVCSKELM [34]	KW-SMOTE-SVM [33]	RFLSTSVM-CIL [77]
Proposed	Yes	No	Yes	Yes	No	Yes	Yes

Table 5: Pair-Wise Win-tie-loss sign test of classification models based on their performance in real world KEEL datasets with Gaussian kernel

Significance	TSVM [10]	UTSVM [21]	RUTSVM-CIL [70]	MVKWELM [34]	MVCSKELM [34]	KW-SMOTE-SVM [33]	RFLSTSVM-CIL [77]
Proposed	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 6: Pair-Wise Win-tie-loss sign test of classification models based on their performance in biomedical datasets with Gaussian kernel

Significance	TSVM [10]	UTSVM [21]	RUTSVM-CIL [70]	MVKWELM [34]	MVCSKELM [34]	KW-SMOTE-SVM [33]	RFLSTSVM-CIL [77]
Proposed	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 7: Statistical comparison of classification models with respect to the proposed KWRUTSVM-CIL on real world KEEL datasets with Gaussian kernel via Wilcoxon signed rank test.

	R+	R-	p-value	Hypothesis (0.05)
TSVM	720	100	0.00001	**Rejected**
UTSVM	623	197	0.00424	**Rejected**
RUTSVM-CIL	674.5	66.5	0.00001	**Rejected**
MVKWELM	820	0	0.00001	**Rejected**
MVCSKELM	806	14	0.00001	**Rejected**
KW-SMOTE-SVM	792.5	27.5	0.00001	**Rejected**
RFLSTSVM-CIL	588	153	0.00164	**Rejected**

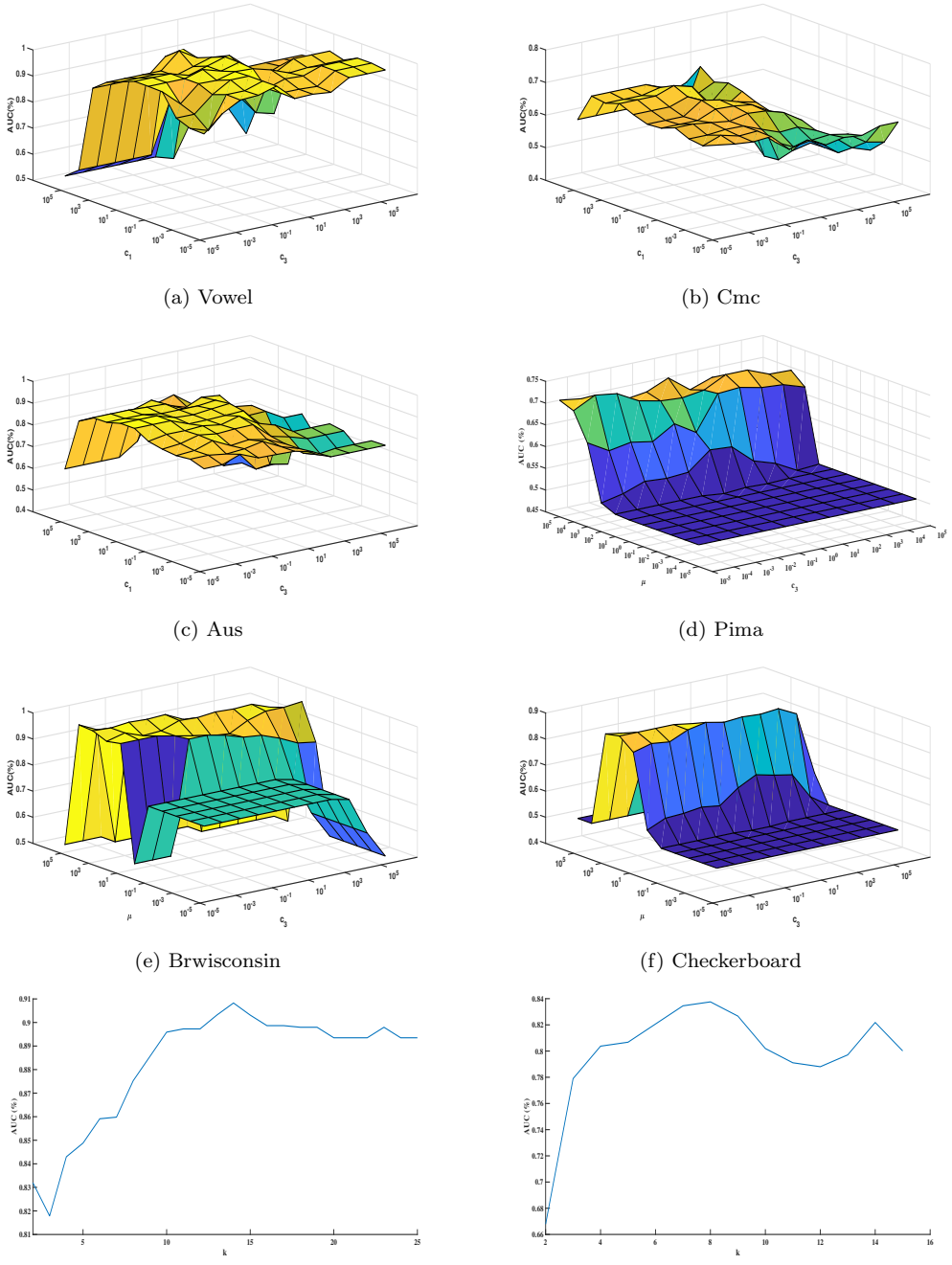
The p-value is calculated from paired Wilcoxon test.
R+ (R-) present the sum of positive (negative) ranks, respectively.

For evaluation of the models via statistical tests, we perform statistical analysis. We used Friedman test with corresponding Nemenyi post hoc test for the comparison of the models. Every classifier in Friedman test is given a rank on a dataset with the worse performing classifier assigned higher rank and vice versa. The average of the rank across different datasets corresponding to a given model is calculated and is known as the average rank of the classifier. The classifier with smallest average rank is the best performing model while as the model with highest rank is the model with lowest performance. Under the null hypothesis, the average rank of the classification models is equal and hence show similar performance. Let N number of datasets are used to evaluate the performance of k number of classification models. The Friedman statistic follows χ^2 distribution with $(k-1)$ degrees of freedom and is given as follows:

$$\chi_F^2 = \frac{12N}{k(k+1)}\Big[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\Big], \tag{59}$$

where R_j is the average rank of the j^{th} classifier. As χ_F^2 is undesirably conservative and thus a better statistic F_F with F-distribution $(k-1, (k-1)(N-1))$ degrees of freedom is given. Mathematically,

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2}. \tag{60}$$



(a) Vowel (b) Cmc

(c) Aus (d) Pima

(e) Brwisconsin (f) Checkerboard

(g) Aus (h) Heart-stat

Figure 1: The sensitivity evaluation of the proposed KWRUTSVM-CIL classifier based on user specified parameters with Gaussian kernel.

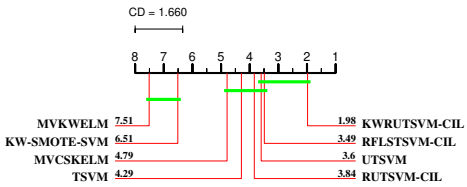


Figure 2: Comparison of the classifiers based on the Nemenyi test. The classifiers which are connected are not significantly different while as the disconnected classifiers have significant difference between them.

After simple calculation with $N = 40$ and $k = 8$, we get $\chi_F^2 = 146.23$ and $F_F = 42.6327$ follows F-distribution with $(7, 273)$ degrees of freedom. For significance level $\alpha = 0.05$, the critical values of $F_F(7, 273)$ is 2.045. Since, $F_F = 42.6327 > 2.045$ hence, null hypothesis gets rejected. Thus, difference is significant among the given classifiers. To check the significant difference between the classifiers, Nemenyi post-hoc test is used. For Nemenyi test, the critical difference (CD) is given by

$$CD = q_\alpha \sqrt{\frac{k \times (k+1)}{6N}}. \tag{61}$$

After simple calculation, with $q_{0.05} = 3.031$ the critical difference is 1.6601. If the average rank of two classifiers differ at least by the CD then the difference between two classifiers is significant with the lower rank classifier performing better than the other classifiers. Table 3 and Table 4 gives the significant difference of the models on KEEL and biomedical datasets, respectively. It is evident that the proposed KWRUTSVM-CIL is performing significantly better than the existing models (except UTSVM and RFLSTSVM-CIL) in KEEL datasets. Figure 2 gives the pictorial representation of the significant difference of the models.

Furthermore, we use Win-Tie-Loss sign test to evaluate the statistical significance of the models. In this test, under null hypothesis two models are equivalent if each of models wins on approximately $N/2$ datasets out of the N datasets. At 5% level of significance, the two models are significantly different if one of models wins approximately on $N/2 + 1.96\sqrt{N}/2$. Also, if there is even number of ties between the two models then the number of ties

are evenly distributed among them otherwise we ignore one and distribute the rest among the given classifiers. Table 5 and Table 6 gives the significant difference between the classifiers based on the pairwise win-tie-loss sign test on KEEL and biomedical datasets, respectively. It is evident that the proposed KWRUTSVM-CIL model is significantly superior compared to the existing models in KEEL and Biomedical datasets.

The nonparametric Wilcoxon signed-rank test [86] is employed to compare the performance of the given models. This test calculates d_i, that is the difference between the accuracy of two compared models corresponding the i-th dataset out of N datasets. The ascending order of absolute differences is ranked wherein the average ranks are given in tie cases. $R+(R-)$ represents the sum of positive (negative) ranks, respectively. The sufficient difference between $R+$ and $R-$ demonstrate that the compared models reject the null hypothesis. The null hypothesis is rejected if the p-value for this test is smaller than 0.05. Therefore, one can see from the Table 7 that the proposed KWRUTSVM-CIL outperforms the compared models, i.e., TWSVM, UTSVM, RUTSVM, MVKWELM, MVCSKELM, KW-SMOTE-SVM and RFLSTSVM-CIL.

From the above analysis, superiority of the proposed KWRUTSM-CIL classifier in comparison with the existing models is evident.

The generalization of the proposed KWRUTSM-CIL is influenced by the hyperparameters. Figure 1 gives the parameter sensitivity of the proposed KWRUTSM-CIL classifier. Thus, the model hyperparameters should be selected carefully for the optimal performance of the proposed KWRUTSM-CIL model.

5. Influence of the K nearest neighbours

Here, we investigate the influence of K nearest neighbours on the performance of the proposed KWRUTSM-CIL classifier. Here, we varied the parameter K from 2 to 10 with step size 1. Table 8 shows the performance of the model with varying number of K nearest neighbours. It is clear from the given table that K needs to be selected carefully for optimal performance.

Table 8: Performance comparison of proposed KWRUTSVM-CIL model with different values of parameter K on real world KEEL datasets with Gaussian kernel.

Datasets	2	3	4	5	6	7	8	9	10
	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean
abalone9-18	0.82 ± 0.05	0.86 ± 0.07	0.87 ± 0.08	0.89 ± 0.04	0.89 ± 0.04	0.88 ± 0.04	0.88 ± 0.04	0.87 ± 0.05	0.87 ± 0.05
	$0.58, 0.61$	$0.58, 0.61$	$0.56, 0.59$	$0.52, 0.57$	$0.48, 0.54$	$0.47, 0.53$	$0.48, 0.54$	$0.46, 0.52$	$0.45, 0.51$
aus	0.8 ± 0.05	0.82 ± 0.05	0.82 ± 0.05	0.83 ± 0.04	0.83 ± 0.04	0.83 ± 0.05	0.85 ± 0.04	0.86 ± 0.05	0.86 ± 0.05
	$0.76, 0.77$	$0.78, 0.79$	$0.79, 0.8$	$0.8, 0.81$	$0.8, 0.81$	$0.81, 0.82$	$0.83, 0.83$	$0.84, 0.84$	$0.85, 0.85$
brwisconsin	0.93 ± 0.02	0.94 ± 0.02	0.95 ± 0.01	0.96 ± 0.01	0.96 ± 0	0.97 ± 0.01	0.97 ± 0.01	0.98 ± 0.01	0.98 ± 0.01
	$0.92, 0.92$	$0.93, 0.93$	$0.94, 0.94$	$0.95, 0.95$	$0.95, 0.95$	$0.95, 0.95$	$0.96, 0.96$	$0.97, 0.97$	$0.97, 0.97$
bupa_or_liver-disorders	0.65 ± 0.07	0.68 ± 0.06	0.68 ± 0.06	0.69 ± 0.08	0.69 ± 0.07	0.68 ± 0.06	0.69 ± 0.06	0.7 ± 0.06	0.71 ± 0.06
	$0.51, 0.54$	$0.59, 0.6$	$0.59, 0.6$	$0.62, 0.62$	$0.61, 0.62$	$0.61, 0.61$	$0.63, 0.64$	$0.64, 0.64$	$0.66, 0.66$
checkerboard	0.8 ± 0.05	0.82 ± 0.05	0.82 ± 0.05	0.83 ± 0.04	0.83 ± 0.04	0.83 ± 0.05	0.85 ± 0.04	0.86 ± 0.05	0.86 ± 0.05
	$0.76, 0.77$	$0.78, 0.79$	$0.79, 0.8$	$0.8, 0.81$	$0.8, 0.81$	$0.81, 0.82$	$0.83, 0.83$	$0.84, 0.84$	$0.85, 0.85$
cleve	0.79 ± 0.05	0.82 ± 0.03	0.84 ± 0.03	0.83 ± 0.04	0.85 ± 0.03	0.85 ± 0.03	0.85 ± 0.04	0.85 ± 0.04	0.85 ± 0.03
	$0.74, 0.76$	$0.79, 0.8$	$0.81, 0.82$	$0.81, 0.81$	$0.83, 0.83$	$0.84, 0.84$	$0.83, 0.83$	$0.83, 0.83$	$0.83, 0.84$
cmc	0.63 ± 0.02	0.67 ± 0	0.67 ± 0.01	0.68 ± 0.01	0.68 ± 0.01	0.68 ± 0.02	0.69 ± 0.01	0.69 ± 0.01	0.69 ± 0.01
	$0.46, 0.51$	$0.55, 0.58$	$0.56, 0.59$	$0.57, 0.6$	$0.57, 0.59$	$0.58, 0.6$	$0.58, 0.61$	$0.59, 0.61$	$0.59, 0.61$
ecoli-0-1_vs_2-3-5	0.88 ± 0.07	0.9 ± 0.08	0.9 ± 0.07	0.9 ± 0.08	0.9 ± 0.08	0.89 ± 0.08	0.89 ± 0.08	0.91 ± 0.06	0.91 ± 0.06
	$0.75, 0.77$	$0.79, 0.81$	$0.77, 0.78$	$0.74, 0.76$	$0.73, 0.74$	$0.71, 0.72$	$0.72, 0.74$	$0.75, 0.76$	$0.74, 0.76$
ecoli-0-1_vs_5	0.94 ± 0.11	0.97 ± 0.05	0.96 ± 0.05	0.94 ± 0.11	0.91 ± 0.11	0.9 ± 0.11	0.91 ± 0.11	0.93 ± 0.11	0.93 ± 0.11
	$0.87, 0.88$	$0.89, 0.89$	$0.86, 0.87$	$0.82, 0.83$	$0.75, 0.76$	$0.74, 0.75$	$0.75, 0.76$	$0.78, 0.8$	$0.78, 0.8$
ecoli-0-1-4-6_vs_5	0.93 ± 0.11	0.93 ± 0.11	0.95 ± 0.06	0.92 ± 0.11	0.92 ± 0.12	0.91 ± 0.11	0.92 ± 0.11	0.92 ± 0.11	0.92 ± 0.11
	$0.74, 0.76$	$0.75, 0.77$	$0.76, 0.78$	$0.68, 0.71$	$0.67, 0.7$	$0.64, 0.67$	$0.66, 0.69$	$0.65, 0.68$	$0.65, 0.68$

Continued on next page

Table 8 – continued from previous page

Datasets	2	3	4	5	6	7	8	9	10
	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean
ecoli-0-1-4-7_vs_2-3-5-6	0.9 ± 0.04	0.9 ± 0.04	0.9 ± 0.05	0.89 ± 0.06	0.9 ± 0.05	0.89 ± 0.07	0.9 ± 0.07	0.88 ± 0.1	0.89 ± 0.11
	0.72, 0.74	0.69, 0.7	0.71, 0.73	0.67, 0.69	0.73, 0.74	0.71, 0.72	0.77, 0.78	0.72, 0.73	0.73, 0.74
ecoli-0-1-4-7_vs_5-6	0.92 ± 0.05	0.93 ± 0.05	0.93 ± 0.05	0.93 ± 0.05	0.93 ± 0.06	0.93 ± 0.06	0.93 ± 0.06	0.91 ± 0.08	0.91 ± 0.07
	0.67, 0.69	0.71, 0.74	0.72, 0.74	0.69, 0.71	0.7, 0.72	0.71, 0.74	0.7, 0.72	0.71, 0.73	0.69, 0.71
ecoli-0-2-3-4_vs_5	0.94 ± 0.11	0.94 ± 0.11	0.99 ± 0.01	0.96 ± 0.06	0.94 ± 0.07	0.94 ± 0.07	0.91 ± 0.11	0.94 ± 0.07	0.91 ± 0.11
	0.85, 0.85	0.85, 0.85	0.89, 0.9	0.85, 0.86	0.84, 0.84	0.84, 0.84	0.8, 0.81	0.84, 0.84	0.79, 0.8
ecoli-0-3-4-6_vs_5	0.92 ± 0.11	0.91 ± 0.11	0.96 ± 0.06	0.94 ± 0.07	0.94 ± 0.07	0.94 ± 0.07	0.96 ± 0.05	0.97 ± 0.06	0.96 ± 0.07
	0.84, 0.85	0.82, 0.83	0.88, 0.89	0.88, 0.88	0.86, 0.87	0.86, 0.87	0.87, 0.88	0.91, 0.91	0.89, 0.89
ecoli-0-3-4-7_vs_5-6	0.87 ± 0.1	0.89 ± 0.07	0.93 ± 0.08	0.88 ± 0.12	0.88 ± 0.11	0.89 ± 0.11	0.89 ± 0.11	0.89 ± 0.1	0.89 ± 0.1
	0.68, 0.69	0.7, 0.72	0.74, 0.76	0.65, 0.67	0.66, 0.68	0.68, 0.7	0.71, 0.72	0.7, 0.71	0.7, 0.71
ecoli-0-4-6_vs_5	0.84 ± 0.11	0.9 ± 0.11	0.95 ± 0.07	0.94 ± 0.08	0.94 ± 0.08	0.94 ± 0.08	0.93 ± 0.08	0.93 ± 0.08	0.93 ± 0.08
	0.76, 0.76	0.86, 0.86	0.92, 0.92	0.9, 0.91	0.87, 0.87	0.86, 0.87	0.84, 0.85	0.81, 0.82	0.81, 0.82
ecoli-0-6-7_vs_3-5	0.91 ± 0.1	0.89 ± 0.11	0.91 ± 0.13	0.91 ± 0.12	0.89 ± 0.11	0.84 ± 0.11	0.87 ± 0.09	0.87 ± 0.09	0.83 ± 0.08
	0.75, 0.78	0.71, 0.73	0.72, 0.74	0.71, 0.73	0.68, 0.71	0.62, 0.65	0.65, 0.67	0.69, 0.71	0.63, 0.65
ecoli-0-6-7_vs_5	0.89 ± 0.07	0.9 ± 0.07	0.92 ± 0.07	0.9 ± 0.07	0.92 ± 0.07	0.89 ± 0.07	0.91 ± 0.08	0.91 ± 0.07	0.88 ± 0.06
	0.71, 0.73	0.73, 0.74	0.75, 0.77	0.73, 0.74	0.71, 0.74	0.7, 0.72	0.71, 0.73	0.69, 0.72	0.65, 0.68
ecoli0137vs26	0.9 ± 0.07	0.91 ± 0.06	0.91 ± 0.06	0.94 ± 0.04	0.94 ± 0.04	0.94 ± 0.03	0.94 ± 0.03	0.94 ± 0.03	0.94 ± 0.03
	0.86, 0.86	0.87, 0.87	0.87, 0.87	0.9, 0.9	0.9, 0.9	0.89, 0.89	0.89, 0.89	0.89, 0.89	0.88, 0.89

Continued on next page

Table 8 – continued from previous page

Datasets	2	3	4	5	6	7	8	9	10
	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean
ecoli01vs5	0.97 ± 0.05 $0.95, 0.95$	0.97 ± 0.05 $0.95, 0.95$	1 ± 0.01 $0.98, 0.98$	1 ± 0.01 $0.98, 0.98$	1 ± 0.01 $0.98, 0.98$	1 ± 0.01 $0.96, 0.96$	1 ± 0.01 $0.96, 0.96$	1 ± 0.01 $0.96, 0.96$	0.99 ± 0.01 $0.94, 0.94$
ecoli2	0.85 ± 0.02 $0.65, 0.66$	0.87 ± 0.05 $0.66, 0.68$	0.89 ± 0.04 $0.65, 0.67$	0.92 ± 0.03 $0.67, 0.7$	0.91 ± 0.03 $0.65, 0.69$	0.91 ± 0.04 $0.64, 0.68$	0.91 ± 0.03 $0.63, 0.67$	0.91 ± 0.03 $0.63, 0.67$	0.91 ± 0.03 $0.64, 0.68$
ecoli3	0.85 ± 0.02 $0.65, 0.66$	0.87 ± 0.05 $0.66, 0.68$	0.89 ± 0.04 $0.65, 0.67$	0.92 ± 0.03 $0.67, 0.7$	0.91 ± 0.03 $0.65, 0.69$	0.91 ± 0.04 $0.64, 0.68$	0.91 ± 0.03 $0.63, 0.67$	0.91 ± 0.03 $0.63, 0.67$	0.91 ± 0.03 $0.64, 0.68$
ecoli4	0.94 ± 0.07 $0.86, 0.87$	0.94 ± 0.07 $0.8, 0.81$	0.98 ± 0.01 $0.79, 0.81$	0.97 ± 0.02 $0.69, 0.73$	0.96 ± 0.02 $0.6, 0.65$	0.94 ± 0.02 $0.52, 0.59$	0.92 ± 0.02 $0.44, 0.53$	0.9 ± 0.02 $0.39, 0.49$	0.89 ± 0.02 $0.36, 0.47$
glass4	0.95 ± 0.07 $0.83, 0.85$	0.96 ± 0.04 $0.7, 0.74$	0.95 ± 0.06 $0.64, 0.69$	0.95 ± 0.06 $0.64, 0.69$	0.95 ± 0.06 $0.64, 0.69$	0.95 ± 0.06 $0.62, 0.67$	0.95 ± 0.06 $0.62, 0.67$	0.95 ± 0.06 $0.62, 0.67$	0.95 ± 0.06 $0.62, 0.67$
heart-stat	0.74 ± 0.03 $0.67, 0.69$	0.77 ± 0.02 $0.72, 0.73$	0.78 ± 0.05 $0.74, 0.75$	0.8 ± 0.03 $0.76, 0.77$	0.81 ± 0.04 $0.77, 0.78$	0.82 ± 0.03 $0.8, 0.8$	0.83 ± 0.04 $0.81, 0.81$	0.82 ± 0.05 $0.79, 0.79$	0.82 ± 0.04 $0.79, 0.8$
iono	0.91 ± 0.04 $0.89, 0.89$	0.93 ± 0.03 $0.91, 0.91$	0.94 ± 0.03 $0.92, 0.92$	0.93 ± 0.03 $0.91, 0.91$	0.93 ± 0.03 $0.91, 0.91$	0.94 ± 0.03 $0.91, 0.91$	0.94 ± 0.02 $0.91, 0.91$	0.94 ± 0.02 $0.91, 0.91$	0.94 ± 0.02 $0.91, 0.91$
new-thyroid1	0.8 ± 0.09	0.9 ± 0.12	0.94 ± 0.09	1 ± 0	1 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.98 ± 0.02
	$0.74, 0.77$	$0.87, 0.89$	$0.93, 0.94$	$1, 1$	$0.99, 0.99$	$0.96, 0.96$	$0.95, 0.95$	$0.94, 0.94$	$0.91, 0.92$
pima	0.7 ± 0.03 $0.58, 0.6$	0.71 ± 0.02 $0.63, 0.63$	0.73 ± 0.05 $0.66, 0.67$	0.71 ± 0.05 $0.65, 0.68$	0.7 ± 0.06 $0.64, 0.67$	0.66 ± 0.04 $0.61, 0.65$	0.66 ± 0.04 $0.61, 0.65$	0.68 ± 0.05 $0.62, 0.66$	0.69 ± 0.05 $0.63, 0.66$
vehicle1	0.77 ± 0.05 $0.65, 0.65$	0.8 ± 0.02 $0.68, 0.68$	0.82 ± 0.02 $0.7, 0.71$	0.84 ± 0.01 $0.72, 0.73$	0.85 ± 0.02 $0.73, 0.74$	0.85 ± 0.03 $0.73, 0.74$	0.84 ± 0.02 $0.72, 0.73$	0.85 ± 0.02 $0.72, 0.74$	0.85 ± 0.03 $0.73, 0.74$
vehicle2	0.97 ± 0.02 $0.96, 0.96$	0.97 ± 0.02 $0.96, 0.96$	0.97 ± 0.02 $0.96, 0.96$	0.97 ± 0.02 $0.96, 0.96$	0.97 ± 0.02 $0.96, 0.96$	0.97 ± 0.02 $0.96, 0.96$	0.97 ± 0.02 $0.96, 0.96$	0.97 ± 0.02 $0.97, 0.97$	0.97 ± 0.02 $0.96, 0.97$
votes	0.95 ± 0.01 $0.93, 0.93$	0.96 ± 0.02 $0.95, 0.95$	0.96 ± 0.03 $0.95, 0.95$	0.96 ± 0.03 $0.95, 0.95$	0.96 ± 0.03 $0.95, 0.95$	0.96 ± 0.02 $0.95, 0.95$	0.96 ± 0.02 $0.95, 0.95$	0.96 ± 0.02 $0.95, 0.95$	0.97 ± 0.02 $0.96, 0.96$
vowel	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	1 ± 0	1 ± 0	1 ± 0

Continued on next page

Table 8 – continued from previous page

Datasets	2	3	4	5	6	7	8	9	10
	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean
	0.99, 0.99	0.99, 0.99	0.98, 0.98	0.99, 0.99	0.98, 0.98	0.98, 0.98	0.98, 0.98	0.98, 0.98	0.99, 0.99
wpbc	0.67 ± 0.1	0.7 ± 0.08	0.68 ± 0.07	0.69 ± 0.07	0.69 ± 0.07	0.7 ± 0.07	0.69 ± 0.08	0.67 ± 0.07	0.65 ± 0.07
	0.5, 0.51	0.53, 0.54	0.51, 0.53	0.52, 0.54	0.52, 0.54	0.53, 0.54	0.51, 0.53	0.49, 0.5	0.47, 0.49
yeast-0-2-5-6_vs_3-7-8-9	0.68 ± 0.04	0.74 ± 0.02	0.77 ± 0.05	0.79 ± 0.03	0.81 ± 0.02	0.81 ± 0.02	0.81 ± 0.02	0.82 ± 0.02	0.82 ± 0.02
	0.5, 0.53	0.58, 0.6	0.62, 0.63	0.63, 0.64	0.65, 0.65	0.64, 0.64	0.62, 0.62	0.64, 0.64	0.63, 0.64
yeast-0-2-5-7-9_vs_3-6-8	0.9 ± 0.04	0.9 ± 0.04	0.9 ± 0.04	0.92 ± 0.04	0.92 ± 0.04	0.92 ± 0.04	0.92 ± 0.04	0.91 ± 0.04	0.91 ± 0.04
	0.77, 0.78	0.77, 0.78	0.78, 0.78	0.79, 0.79	0.78, 0.79	0.78, 0.79	0.77, 0.78	0.74, 0.75	0.73, 0.74
yeast-0-5-6-7-9_vs_4	0.79 ± 0.13	0.82 ± 0.12	0.82 ± 0.1	0.81 ± 0.09	0.83 ± 0.11	0.82 ± 0.1	0.79 ± 0.08	0.78 ± 0.08	0.79 ± 0.09
	0.59, 0.6	0.58, 0.59	0.56, 0.58	0.53, 0.56	0.57, 0.59	0.51, 0.54	0.47, 0.5	0.44, 0.48	0.45, 0.49
yeast-2_vs_4	0.88 ± 0.05	0.88 ± 0.04	0.88 ± 0.04	0.88 ± 0.04	0.89 ± 0.03	0.88 ± 0.03	0.89 ± 0.02	0.88 ± 0.03	0.87 ± 0.04
	0.81, 0.81	0.8, 0.8	0.78, 0.78	0.78, 0.79	0.77, 0.77	0.75, 0.75	0.73, 0.73	0.71, 0.72	0.71, 0.71
yeast2vs8	0.77 ± 0.16	0.77 ± 0.16	0.8 ± 0.14	0.77 ± 0.16	0.77 ± 0.16	0.77 ± 0.16	0.77 ± 0.16	0.77 ± 0.16	0.77 ± 0.16
	0.65, 0.69	0.65, 0.69	0.7, 0.73	0.63, 0.66	0.65, 0.69	0.63, 0.66	0.65, 0.69	0.65, 0.69	0.63, 0.66
yeast3	0.9 ± 0.02	0.91 ± 0.02	0.9 ± 0.02	0.91 ± 0.02	0.92 ± 0.02	0.92 ± 0.02	0.93 ± 0.02	0.93 ± 0.03	0.93 ± 0.02
	0.77, 0.77	0.77, 0.77	0.75, 0.76	0.75, 0.76	0.74, 0.76	0.75, 0.76	0.74, 0.76	0.74, 0.75	0.73, 0.75
yeast5	0.95 ± 0.03	0.98 ± 0.01	0.98 ± 0.01	0.98 ± 0.01	0.98 ± 0.01	0.98 ± 0.01	0.98 ± 0.01	0.98 ± 0.01	0.98 ± 0.01
	0.6, 0.64	0.6, 0.66	0.59, 0.65	0.62, 0.67	0.61, 0.66	0.6, 0.66	0.61, 0.67	0.6, 0.66	0.6, 0.65

Table 9: Biomedical dataset specifications

	Subclasses	Sample Size	Features
Alzheimer's disease dataset	Control Normal (CN)	228	91
	Mild Cognitive Impairment (MCI)	398	91
	Alzheimer's disease (AD)	187	91
Breast cancer dataset	Adenosis (ADN)	106	768
	Ductal Carcinoma (DC)	208	768
	Fibroadenema (FA)	237	768
	Mucinous Carcinoma (MC)	169	768
	Phyllodes Tumour (PT)	115	768
	Tubular Adenoma (TA)	130	768
	Lobular Carcinoma (LC)	137	768
	Papillary Carcinoma (PC)	138	768

5.1. Biomedical Data

For evaluating the applicability of the classifiers in real world scenarios, we use Alzheimer's disease and breast cancer datasets. Table 9 gives the details of the biomedical data. The performance of the models on biomedical datasets is presented in Table 10. In PT_vs_PC subjects, TA_vs_LC subjects and ADN_vs_DC subjects the proposed model showed the average accuracy of $0.68, 0.76$ and 0.89, respectively which is better in comparison with the existing classifiers. Thus, the proposed KWRUTSVM-CIL model shows better or competitive performance compared to the baseline models. It is evident that the proposed KWRUTSVM-CIL classifier obtained highest average classification performance in terms of accuracy. Also, the average rank is lowest i.e. the proposed model KWRUTSVM-CIL wins in most of the datasets. Hence, better generalization performance is demonstrated by the proposed KWRUTSVM-CIL model which shows its applicability in biomedical domain.

Table 10: Classification accuracy of TSVM, UTSVM, RUTSVM-CIL and proposed KWRUTSVM-CIL models on biomedical datasets with Gaussian kernel.

Datasets	TSVM [10]	UTSVM [21]	RUTSVM-CIL [70]	MVKWELM [34]	MVCSKELM [34]	KW-SMOTE-SVM [33]	RFLSTSVM-CIL [77]	KWRUTSVM-CIL
	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean
ADN_vs_DC	0.86 ± 0.05 $0.8, 0.81$	0.88 ± 0.07 $0.83, 0.84$	0.87 ± 0.03 $0.8, 0.81$	0.86 ± 0.03 $0.8, 0.81$	0.87 ± 0.03 $0.8, 0.82$	0.87 ± 0.04 $0.81, 0.82$	0.86 ± 0.06 $0.81, 0.82$	$\mathbf{0.89} \pm 0.02$ $0.83, 0.83$
ADN_vs_LC	0.57 ± 0.04 $0.53, 0.53$	$\mathbf{0.62} \pm 0.08$ $0.66, 0.68$	0.57 ± 0.07 $0.39, 0.42$	0.52 ± 0.07 $0.39, 0.4$	0.57 ± 0.05 $0.63, 0.67$	0.56 ± 0.06 $0.51, 0.52$	0.56 ± 0.05 $0.5, 0.51$	0.61 ± 0.09 $0.54, 0.54$
ADN_vs_MC	0.63 ± 0.05 $0.63, 0.67$	0.62 ± 0.03 $0.6, 0.62$	0.61 ± 0.06 $0.58, 0.6$	0.58 ± 0.08 $0.55, 0.57$	0.63 ± 0.04 $0.62, 0.66$	0.6 ± 0.09 $0.53, 0.53$	0.62 ± 0.06 $0.59, 0.61$	$\mathbf{0.64} \pm 0.04$ $0.62, 0.66$
ADN_vs_PC	0.69 ± 0.08 $0.69, 0.71$	0.68 ± 0.03 $0.69, 0.7$	0.68 ± 0.08 $0.7, 0.73$	0.49 ± 0.06 $0.54, 0.56$	0.6 ± 0.1 $0.61, 0.62$	0.56 ± 0.08 $0.52, 0.52$	0.68 ± 0.08 $0.68, 0.7$	$\mathbf{0.73} \pm 0.07$ $0.71, 0.71$
CN_vs_AD	$\mathbf{0.89} \pm 0.05$ $0.88, 0.88$	0.88 ± 0.02 $0.87, 0.87$	0.88 ± 0.04 $0.87, 0.87$	0.86 ± 0.04 $0.84, 0.84$	0.88 ± 0.04 $0.86, 0.86$	0.87 ± 0.03 $0.85, 0.85$	$\mathbf{0.89} \pm 0.05$ $0.87, 0.88$	0.87 ± 0.05 $0.86, 0.86$
CN_vs_MCI	0.69 ± 0.06 $0.63, 0.64$	0.69 ± 0.03 $0.63, 0.65$	0.69 ± 0.01 $0.62, 0.63$	0.67 ± 0.01 $0.61, 0.62$	0.69 ± 0.02 $0.64, 0.66$	0.69 ± 0.03 $0.62, 0.63$	0.69 ± 0.03 $0.62, 0.63$	$\mathbf{0.7} \pm 0.03$ $0.64, 0.66$
FA_vs_DC	0.81 ± 0.01 $0.81, 0.81$	0.82 ± 0.03 $0.81, 0.81$	0.8 ± 0.03 $0.8, 0.8$	0.81 ± 0.02 $0.79, 0.79$	$\mathbf{0.83} \pm 0.01$ $0.82, 0.82$	0.82 ± 0.02 $0.82, 0.82$	0.81 ± 0.03 $0.81, 0.81$	0.82 ± 0.02 $0.82, 0.82$
FA_vs_LC	0.64 ± 0.06 $0.54, 0.54$	0.64 ± 0.05 $0.52, 0.52$	$\mathbf{0.66} \pm 0.04$ $0.56, 0.56$	0.62 ± 0.04 $0.58, 0.6$	0.6 ± 0.04 $0.52, 0.52$	0.55 ± 0.05 $0.46, 0.46$	0.63 ± 0.05 $0.5, 0.51$	0.65 ± 0.03 $0.55, 0.55$
FA_vs_MC	0.55 ± 0.02 $0.36, 0.4$	0.58 ± 0.08 $0.48, 0.48$	0.55 ± 0.03 $0.39, 0.41$	0.58 ± 0.05 $0.46, 0.47$	$\mathbf{0.59} \pm 0.06$ $0.5, 0.5$	0.51 ± 0.05 $0.4, 0.4$	0.57 ± 0.07 $0.48, 0.49$	0.58 ± 0.07 $0.52, 0.52$
FA_vs_PC	0.63 ± 0.05 $0.54, 0.54$	0.63 ± 0.04 $0.54, 0.55$	0.56 ± 0.07 $0.47, 0.48$	0.51 ± 0.08 $0.37, 0.37$	0.58 ± 0.04 $0.35, 0.39$	0.58 ± 0.04 $0.4, 0.41$	$\mathbf{0.65} \pm 0.08$ $0.54, 0.55$	$\mathbf{0.65} \pm 0.02$ $0.54, 0.55$
MCI_vs_AD	0.68 ± 0.06 $0.58, 0.59$	0.68 ± 0.05 $0.57, 0.57$	$\mathbf{0.69} \pm 0.04$ $0.59, 0.6$	$\mathbf{0.69} \pm 0.05$ $0.59, 0.6$	0.66 ± 0.05 $0.54, 0.55$	0.68 ± 0.05 $0.58, 0.59$	$\mathbf{0.69} \pm 0.05$ $0.6, 0.62$	$\mathbf{0.69} \pm 0.04$ $0.59, 0.6$

Continued on next page

Table 10 – continued from previous page

Datasets	TSVM [10]	UTSVM [21]	RUTSVM-CIL [70]	MVKWELM [34]	MVCSKELM [34]	KW-SMOTE-SVM [33]	RFLSTSVM-CIL [77]	KWRUTSVM-CIL
	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean	AUC±STD F-Measure, G-Mean
PT_vs_DC	0.85 ± 0.04 $0.81, 0.81$	0.86 ± 0.04 $0.81, 0.82$	0.87 ± 0.06 $0.82, 0.83$	0.87 ± 0.05 $0.82, 0.82$	$\mathbf{0.88} \pm 0.05$ $0.84, 0.84$	$\mathbf{0.88} \pm 0.04$ $0.84, 0.84$	0.87 ± 0.05 $0.82, 0.83$	$\mathbf{0.88} \pm 0.05$ $0.84, 0.84$
PT_vs_MC	0.6 ± 0.04 $0.61, 0.63$	$\mathbf{0.62} \pm 0.05$ $0.62, 0.65$	0.58 ± 0.08 $0.42, 0.44$	0.59 ± 0.04 $0.59, 0.6$	0.61 ± 0.03 $0.61, 0.63$	0.54 ± 0.03 $0.49, 0.5$	0.6 ± 0.07 $0.54, 0.55$	0.61 ± 0.07 $0.57, 0.58$
PT_vs_PC	0.65 ± 0.04 $0.66, 0.67$	0.65 ± 0.03 $0.66, 0.67$	0.66 ± 0.08 $0.67, 0.68$	0.58 ± 0.06 $0.63, 0.65$	0.61 ± 0.02 $0.63, 0.65$	0.54 ± 0.04 $0.53, 0.53$	0.63 ± 0.07 $0.64, 0.64$	$\mathbf{0.68} \pm 0.05$ $0.65, 0.65$
TA_vs_DC	0.75 ± 0.02 $0.69, 0.69$	0.75 ± 0.04 $0.69, 0.69$	0.69 ± 0.07 $0.56, 0.6$	0.73 ± 0.03 $0.65, 0.67$	0.75 ± 0.03 $0.68, 0.69$	0.75 ± 0.04 $0.69, 0.69$	0.75 ± 0.03 $0.67, 0.69$	$\mathbf{0.76} \pm 0.04$ $0.71, 0.72$
TA_vs_LC	0.75 ± 0.06 $0.75, 0.76$	0.75 ± 0.06 $0.75, 0.75$	0.74 ± 0.06 $0.77, 0.78$	0.64 ± 0.02 $0.55, 0.57$	0.7 ± 0.07 $0.64, 0.65$	0.56 ± 0.06 $0.54, 0.54$	0.75 ± 0.04 $0.75, 0.75$	$\mathbf{0.76} \pm 0.04$ $0.77, 0.78$
TA_vs_MC	0.63 ± 0.06 $0.58, 0.58$	0.63 ± 0.05 $0.58, 0.58$	0.6 ± 0.03 $0.51, 0.52$	0.59 ± 0.02 $0.45, 0.47$	0.62 ± 0.04 $0.41, 0.48$	0.57 ± 0.04 $0.5, 0.5$	$\mathbf{0.65} \pm 0.07$ $0.61, 0.61$	0.64 ± 0.03 $0.59, 0.59$
Average AUC	0.6988	0.7052	0.6877	0.6579	0.6867	0.6547	0.7001	**0.716**
Average Rank	4.3235	3.4706	5.1176	6.7059	4.1176	6.2353	4.3235	**1.7059**
Overall Win-Tie-Loss	[1,0,1]	[2,0,0]	[1,0,2]	[0,0,6]	[2,0,1]	[0,0,6]	[2,0,1]	**[9,0,0]**

6. Conclusion

In this paper, we proposed a novel K-nearest neighbour weighted reduced universum twin support vector machines for class imbalance learning (KWRUTSVM-CIL). To create a balance in the classes, the proposed KWRUTSVM-CIL model uses universum data. Also, the proposed model exploits the local neighbourhood information via incorporation of weight matrix in its objective function. To exploit the inter-class information, the weight vector is incorporated in the constraints of the corresponding objective functions. Different from TSVM, UTSVM and RUTSVM-CIL, we added an extra regularisation term in the objective function of the proposed KWRUTSVM-CIL formulation for maximising the margin and ensuring that the matrices in the Wolfe dual of the proposed KWRUTSVM-CIL formulation are positive definite and structural risk minimization principle is implemented which is the marrow of statistical learning. Combination of undersampling with oversampling using universum data leads to the improved results in the class imbalance problems. The efficacy of the proposed KWRUTSVM-CIL model is demonstrated by the experimental results and the statistical analysis on benchmark KEEL and UCI datasets. As an application, we used the proposed KWRUTSVM-CIL classifier for the diagnosis of diseases in biomedical domain like Alzheimer's disease and breast cancer disease. The proposed model showed highest average accuracy and lowest rank in comparison with the baseline classifiers in biomedical datasets. In future, one can explore the selection of universum data as it impacts the performance of the model.

Acknowledgment

This work is funded by Government of India, Science and Engineering Research Board (SERB) under Ramanujan Fellowship Scheme, Grant No. SB/S2/RJN-001/2016, Department of Science and Technology under Interdisciplinary Cyber Physical Systems (ICPS) Scheme grant no. DST/ICPS/CPS-Individual/2018/276 and National Supercomputing Mission under DST and Miety, Govt. of India under Grant No. DST/NSM/R&D_HPC_Appl/2021/03.29. We acknowledge Council of Scientific & Industrial Research (CSIR), New Delhi, INDIA for Extra Mural Research (EMR) grant, under grant no. 22(0751)/17/EMR-II. We are grateful for the facilities and support by Indian Institute of Technology Indore. The data used in this study was funded by the ADNI (National Institutes of Health Grant U01 AG024904), and DOD ADNI (Department of Defense award number W81XWH-12–2-0012) whose funding is

provided by National Institute on Aging, the 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.;

Alzheimer's Drug Discovery Foundation; Araclon Biotech; BioClinica, Inc.; Biogen; Bristol-Myers Squibb Company; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Merck & Co., Inc.; 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 & Development LLC.; Eli Lilly and Company; Lumosity; Lundbeck; Neurotrack Technologies; Takeda Pharmaceutical Company; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; and Transition Therapeutics. The Canadian Institutes of Health Research is providing funds to sup-port ADNI clinical sites in Canada. Foundation for the National Institutes of Health (www.fnih.org) provided private sector contributions. The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Therapeutic Research Institute at the University of Southern California. The dissemination of ADNI data is carried out by the Laboratory for Neuro Imaging at the University of Southern California.

CRediT authorship contribution statement

M.A. Ganaie: Conceptualization; Formal analysis; Investigation; Methodology; Software; Validation; Visualization; Writing - original draft; Writing - review & editing. **M. Tanveer:** Conceptualization; Methodology; Validation; Supervision; Funding acquisition; Project administration; Resources; Writing–Review & editing.

References

[1] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[2] Xiaochen Zhang, Dongxiang Jiang, Te Han, Nanfei Wang, Wenguang Yang, and Yizhou Yang. Rotating machinery fault diagnosis for imbalanced data based on fast clustering algorithm and support vector machine. *Journal of Sensors*, 2017, 2017.

[3] Bharat Richhariya and M. Tanveer. EEG signal classification using universum support vector machine. *Expert Systems with Applications*, 106:169–182, 2018.

[4] Bharat Richhariya, M. Tanveer, AH Rashid, and Alzheimer's Disease Neuroimaging Initiative. Diagnosis of alzheimer's disease using universum support vector machine based recursive feature elimination (USVM-RFE). *Biomedical Signal Processing and Control*, 59:101903, 2020.

[5] Lixin Shen, Hong Wang, Li Da Xu, Xue Ma, Sohail Chaudhry, and Wu He. Identity management based on PCA and SVM. *Information Systems Frontiers*, 18(4):711–716, 2016.

[6] Bharat Richhariya and Deepak Gupta. Facial expression recognition using iterative universum twin support vector machine. *Applied Soft Computing*, 76:53–67, 2019.

[7] Ruixi Yuan, Zhu Li, Xiaohong Guan, and Li Xu. An SVM-based machine learning method for accurate internet traffic classification. *Information Systems Frontiers*, 12(2):149–156, 2010.

[8] Xuhui Wang, Xinfeng Chen, and Zhuming Bi. Support vector machine and ROC curves for modeling of aircraft fuel consumption. *Journal of Management Analytics*, 2(1):22–34, 2015.

[9] Donghong Ding, Fengyang He, Lei Yuan, Zengxi Pan, Lei Wang, and Montserrat Ros. The first step towards intelligent wire arc additive manufacturing: An automatic bead modelling system using machine learning through industrial information integration. *Journal of Industrial Information Integration*, 23:100218, 2021.

[10] Jayadeva, Reshma Khemchandani, and Suresh Chandra. Twin support vector machines for pattern classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):905–910, 2007.

[11] M Arun Kumar and Madan Gopal. Least squares twin support vector machines for pattern classification. *Expert Systems with Applications*, 36(4):7535–7543, 2009.

[12] Xinjun Peng. A ν-twin support vector machine (ν-TSVM) classifier and its geometric algorithms. *Information Sciences*, 180(20):3863–3875, 2010.

[13] Chunyan Wang, Qiaolin Ye, Peng Luo, Ning Ye, and Liyong Fu. Robust capped L_1-norm twin support vector machine. *Neural Networks*, 114: 47–59, 2019.

[14] He Yan, Qiaolin Ye, Tian'an Zhang, Dong-Jun Yu, Xia Yuan, Yiqing Xu, and Liyong Fu. Least squares twin bounded support vector machines based on L_1-norm distance metric for classification. *Pattern Recognition*, 74:434–447, 2018.

[15] Henghao Zhao, Qiaolin Ye, Meen Abdullah Naiem, and Liyong Fu. Robust $l\_\{2,1\}$-Norm distance enhanced multi-weight vector projection support vector machine. *IEEE Access*, 7:3275–3286, 2018.

[16] Ramin Rezvani-KhorashadiZadeh and Monsefi Reza. WS-TWSVM: weighted structural twin support vector machine by local and global information. In *2015 5th International Conference on Computer and Knowledge Engineering (ICCKE)*, pages 170–175. IEEE, 2015.

[17] Shiliang Sun, Xijiong Xie, and Chao Dong. Multiview learning with generalized eigenvalue proximal support vector machines. *IEEE Transactions on Cybernetics*, 49(2):688–697, 2018.

[18] Xijiong Xie. Regularized multi-view least squares twin support vector machines. *Applied Intelligence*, 48(9):3108–3115, 2018.

[19] Shifei Ding, Xiekai Zhang, Yuexuan An, and Yu Xue. Weighted linear loss multiple birth support vector machine based on information granulation for multi-class classification. *Pattern Recognition*, 67:32–46, 2017.

[20] Jason Weston, Ronan Collobert, Fabian Sinz, Léon Bottou, and Vladimir Vapnik. Inference with the universum. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 1009–1016, 2006.

[21] Zhiquan Qi, Yingjie Tian, and Yong Shi. Twin support vector machine with universum data. *Neural Networks*, 36:112–119, 2012.

[22] Yitian Xu, Mei Chen, and Guohui Li. Least squares twin support vector machine with universum data for classification. *International Journal of Systems Science*, 47(15):3637–3645, 2016.

[23] B Richhariya and M. Tanveer. A fuzzy universum least squares twin support vector machine (FULSTSVM). *Neural Computing and Applications*, pages 1–12, 2021. doi: 10.1007/s00521-021-05721-4.

[24] Fabian H Sinz, Olivier Chapelle, Alekh Agarwal, and Bernhard Schölkopf. An analysis of inference with the universum. In *NIPS*, volume 7, pages 1–1, 2007.

[25] Divya Tomar and Sonali Agarwal. Hybrid feature selection based weighted least squares twin support vector machine approach for diagnosing breast cancer, hepatitis, and diabetes. *Advances in Artificial Neural Systems*, 2015, 2015.

[26] Rukshan Batuwita and Vasile Palade. FSVM-CIL: fuzzy support vector machines for class imbalance learning. *IEEE Transactions on Fuzzy Systems*, 18(3):558–571, 2010.

[27] Kai Li and Hongyan Ma. A fuzzy twin support vector machine algorithm. *International Journal of Application or Innovation in Engineering and Management (IJAIEM)*, 2(3):459–465, 2013.

[28] Benjamin X Wang and Nathalie Japkowicz. Boosting support vector machines for imbalanced data sets. *Knowledge and Information Systems*, 25(1):1–20, 2010.

[29] Qi Fan, Zhe Wang, Dongdong Li, Daqi Gao, and Hongyuan Zha. Entropy-based fuzzy support vector machine for imbalanced datasets. *Knowledge-Based Systems*, 115:87–99, 2017.

[30] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[31] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550, 2008.

[32] Dong-Jun Yu, Jun Hu, Zhen-Min Tang, Hong-Bin Shen, Jian Yang, and Jing-Yu Yang. Improving protein-ATP binding residues prediction by boosting SVMs with random under-sampling. *Neurocomputing*, 104: 180–190, 2013.

[33] Josey Mathew, Chee Khiang Pang, Ming Luo, and Weng Hoe Leong. Classification of imbalanced data by oversampling in kernel space of support vector machines. *IEEE Transactions on Neural Networks and Learning Systems*, 29(9):4065–4076, 2017.

[34] Bhagat Singh Raghuwanshi and Sanyam Shukla. Minimum variance-embedded kernelized extension of extreme learning machine for imbalance learning. *Pattern Recognition*, page 108069, 2021.

[35] Yuchun Tang, Yan-Qing Zhang, Nitesh V Chawla, and Sven Krasser. SVMs modeling for highly imbalanced classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(1):281–288, 2008.

[36] M. Tanveer, A Sharma, and P. N. Suganthan. General twin support vector machine with pinball loss function. *Information Sciences*, 494: 311–327, 2019.

[37] M. A. Ganaie and M. Tanveer. Robust general twin support vector machine with pinball loss function. In *Machine Learning for Intelligent Multimedia Analytics*, pages 103–125. Springer, 2021.

[38] M. Tanveer, A. Tiwari, R. Choudhary, and M. A. Ganaie. Large-scale pinball twin support vector machines. *Machine Learning*, pages 1–24, 2021. doi: 10.1007/s10994-021-06061-z.

[39] Yitian Xu, Zhiji Yang, Yuqun Zhang, Xianli Pan, and Laisheng Wang. A maximum margin and minimum volume hyper-spheres machine with pinball loss for imbalanced data classification. *Knowledge-Based Systems*, 95:75–85, 2016.

[40] Yitian Xu, Yuqun Zhang, Jiang Zhao, Zhiji Yang, and Xianli Pan. KNN-based maximum margin and minimum volume hyper-sphere machine for imbalanced data classification. *International Journal of Machine Learning and Cybernetics*, 10(2):357–368, 2019.

[41] M. A. Ganaie, M. Tanveer, and Iman Beheshti. Brain age prediction using improved twin SVR. *Neural Computing and Applications*, pages 1–11, 2022. doi: 10.1007/s00521-021-06518-1.

[42] M. A. Ganaie, M. Tanveer, and Iman Beheshti. Brain age prediction with improved least squares twin SVR. *IEEE Journal of Biomedical and Health Informatics*, 2022. doi: 10.1109/JBHI.2022.3147524.

[43] Iman Beheshti, M. A. Ganaie, Vardhan Paliwal, Aryan Rastogi, Imran Razzak, and M Tanveer. Predicting brain age using machine learning algorithms: A comprehensive evaluation. *IEEE Journal of Biomedical and Health Informatics*, 2021. doi: 10.1109/JBHI.2021.3083187.

[44] Yuh-Jye Lee and Olvi L Mangasarian. RSVM: Reduced support vector machines. In *Proceedings of the 2001 SIAM International Conference on Data Mining*, pages 1–17. SIAM, 2001.

[45] Mittul Singh, Jivitej Chadha, Puneet Ahuja, and Suresh Chandra. Reduced twin support vector regression. *Neurocomputing*, 74(9):1474–1477, 2011.

[46] Yuh-Jye Lee and Su-Yun Huang. Reduced support vector machines: A statistical theory. *IEEE Transactions on Neural Networks*, 18(1):1–13, 2007.

[47] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[48] Qiaolin Ye, Chunxia Zhao, Shangbing Gao, and Hao Zheng. Weighted twin support vector machines with local information and its application. *Neural Networks*, 35:31–39, 2012.

[49] Yitian Xu, Jia Yu, and Yuqun Zhang. KNN-based weighted rough ν-twin support vector machine. *Knowledge-Based Systems*, 71:303–313, 2014.

[50] Yitian Xu and Laisheng Wang. K-nearest neighbor-based weighted twin support vector regression. *Applied Intelligence*, 41(1):299–309, 2014.

[51] Yitian Xu. K-nearest neighbor-based weighted multi-class twin support vector machine. *Neurocomputing*, 205:430–438, 2016.

[52] Xianli Pan, Yao Luo, and Yitian Xu. K-nearest neighbor based structural twin support vector machine. *Knowledge-Based Systems*, 88:34–44, 2015.

[53] M. Tanveer, A. Sharma, and P. N. Suganthan. Least squares KNN-based weighted multiclass twin SVM. *Neurocomputing*, 459:454–464, 2021.

[54] M. A. Ganaie, Minghui Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan. Ensemble deep learning: A review. *arXiv preprint arXiv:2104.02395*, 2021.

[55] M. A. Ganaie, M. Tanveer, and P. N. Suganthan. Oblique decision tree ensemble via twin bounded SVM. *Expert Systems with Applications*, 143:113072, 2020.

[56] M. A. Ganaie and M. Tanveer. LSTSVM classifier with enhanced features from pre-trained functional link network. *Applied Soft Computing*, 93:106305, 2020.

[57] M. Tanveer, M. A. Ganaie, and P. N. Suganthan. Ensemble of classification models with weighted functional link network. *Applied Soft Computing*, 107:107322, 2021.

[58] M. Tanveer, T. Rajani, R. Rastogi, Y. H. Shao, and M. A. Ganaie. Comprehensive review on twin support vector machines. *Annals of Operations Research*, 2022. doi: 10.1007/s10479-022-04575-w.

[59] B. Richhariya and M. Tanveer. An efficient angle-based universum least squares twin support vector machine for classification. *ACM Transactions on Internet Technology*, 21(3), 2021. doi: 10.1145/3387131.

[60] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[61] Hamid Parvin, Behrouz Minaei-Bidgoli, and Hosein Alizadeh. Detection of cancer patients using an innovative method for learning at imbalanced datasets. In *International conference on rough sets and knowledge technology*, pages 376–381. Springer, 2011.

[62] Shuo Wang and Xin Yao. Using class imbalance learning for software defect prediction. *IEEE Transactions on Reliability*, 62(2):434–443, 2013.

[63] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

[64] Miroslav Kubat, Robert C Holte, and Stan Matwin. Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, 30(2):195–215, 1998.

[65] Bartosz Krawczyk, Mikel Galar, Łukasz Jeleń, and Francisco Herrera. Evolutionary undersampling boosting for imbalanced classification of breast cancer malignancy. *Applied Soft Computing*, 38:714–726, 2016.

[66] M. A. Ganaie, M. Tanveer, and Alzheimer's Disease Neuroimaging Initiative. Fuzzy least squares projection twin support vector machines for class imbalance learning. *Applied Soft Computing*, 113:107933, 2021.

[67] Swagatam Das, Shounak Datta, and Bidyut B Chaudhuri. Handling data irregularities in classification: Foundations, trends, and future challenges. *Pattern Recognition*, 81:674–693, 2018.

[68] Kaixiang Yang, Zhiwen Yu, CL Philip Chen, Wenming Cao, Hau-San Wong, Jane You, and Guoqiang Han. Progressive hybrid classifier ensemble for imbalanced data. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021.

[69] Michał Koziarski. Radial-based undersampling for imbalanced data classification. *Pattern Recognition*, 102:107262, 2020.

[70] Bharat Richhariya and M. Tanveer. A reduced universum twin support vector machine for class imbalance learning. *Pattern Recognition*, 102: 107150, 2020.

[71] Iman Nekooeimehr and Susana K Lai-Yuen. Adaptive semi-unsupervised weighted oversampling (A-SUWO) for imbalanced datasets. *Expert Systems with Applications*, 46:405–416, 2016.

[72] Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):63–77, 2005.

[73] Xulei Yang, Qing Song, and Yue Wang. A weighted support vector machine for data classification. *International Journal of Pattern Recognition and Artificial Intelligence*, 21(05):961–976, 2007.

[74] Yuan-Hai Shao, Wei-Jie Chen, Jing-Jing Zhang, Zhen Wang, and Nai-Yang Deng. An efficient weighted lagrangian twin support vector machine for imbalanced data classification. *Pattern Recognition*, 47(9): 3158–3167, 2014.

[75] Salim Rezvani and Xizhao Wang. Class imbalance learning using fuzzy art and intuitionistic fuzzy twin support vector machines. *Information Sciences*, 578:659–682, 2021.

[76] C Jimenez-Castaño, A Alvarez-Meza, and A Orozco-Gutierrez. Enhanced automatic twin support vector machine for imbalanced data classification. *Pattern Recognition*, 107:107442, 2020.

[77] Bharat Richhariya and M. Tanveer. A robust fuzzy least squares twin support vector machine for class imbalance learning. *Applied Soft Computing*, 71:418–432, 2018.

[78] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

[79] Jesús Alcalá-Fdez, Alberto Fernández, Julián Luengo, Joaquín Derrac, Salvador García, Luciano Sánchez, and Francisco Herrera. Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic & Soft Computing*, 17, 2011.

[80] Chunfeng Lian, Mingxia Liu, Jun Zhang, and Dinggang Shen. Hierarchical fully convolutional network for joint atrophy localization and Alzheimer's disease diagnosis using structural MRI. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):880–893, 2018.

[81] Raymond Y Lo and William J Jagust. Predicting missing biomarker data in a longitudinal study of Alzheimer disease. *Neurology*, 78(18): 1376–1382, 2012.

[82] Eric Westman, J-Sebastian Muehlboeck, and Andrew Simmons. Combining MRI and CSF measures for classification of Alzheimer's disease

and prediction of mild cognitive impairment conversion. *Neuroimage*, 62(1):229–238, 2012.

[83] Martin Reuter, Nicholas J Schmansky, H Diana Rosas, and Bruce Fischl. Within-subject template estimation for unbiased longitudinal image analysis. *Neuroimage*, 61(4):1402–1418, 2012.

[84] Fabio A Spanhol, Luiz S Oliveira, Caroline Petitjean, and Laurent Heutte. A dataset for breast cancer histopathological image classification. *IEEE Transactions on Biomedical Engineering*, 63(7):1455–1462, 2015.

[85] Chandan Gautam, Pratik K Mishra, Aruna Tiwari, Bharat Richhariya, Hari Mohan Pandey, Shuihua Wang, M. Tanveer, and Alzheimer's Disease Neuroimaging Initiative. Minimum variance-embedded deep kernel regularized least squares method for one-class classification and its applications to biomedical data. *Neural Networks*, 123:191–216, 2020.

[86] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.