Minimum Variance-Embedded Deep Kernel Regularized least squares Method for One-class Classification and Its Applications to Biomedical Data

Chandan Gautam[a,*], Pratik K Mishra[a], Aruna Tiwari[a], Bharat Richhariya[b], Hari Mohan Pandey[c], Shuihua Wang[d], M. Tanveer[b,*], for the Alzheimer's Disease Neuroimaging Initiative[d]

[a]*Discipline of Computer Science and Engineering, Indian Institute of Technology Indore, Simrol, Indore, 453552, India*
[b]*Discipline of Mathematics, Indian Institute of Technology Indore, Simrol, Indore, 453552, India*
[c]*Department of Computer Science, Edge Hill University, Lancashire, UK*
[d]*School of Architecture Building and Civil engineering, Loughborough University, Loughborough, LE11 3TU, UK*

Abstract

Deep kernel learning has been well explored for multi-class classification tasks; however, relatively less work is done for one-class classification (OCC). OCC needs samples from only one class to train the model. Most recently, kernel regularized least squares (KRL) method-based deep architecture is developed for the OCC task. This paper introduces a novel extension of this method by embedding minimum variance information within this architecture. This embedding improves the generalization capability of the classifier by reducing the intra-class variance. In contrast to traditional deep learning methods, this method can effectively work with small-size datasets. We conduct a comprehensive set of experiments on 18 benchmark datasets (13 biomedical and 5 other datasets) to demonstrate the performance of the proposed classifier. We compare the results with 16 state-of-the-art one-class classifiers. Further, we also test our method for 2 real-world biomedical datasets viz.; detection of Alzheimer's disease from structural magnetic resonance imaging data and detection of breast cancer from histopathological images. Proposed method exhibits more than 5% F_1 score compared to existing state-of-the-art methods for various biomedical benchmark datasets. This makes it viable for application in biomedical fields where relatively less amount of data is available.

The source code is available on the corresponding author's GitHub homepage: `https://github.com/Chandan-IITI/Deep-Kernel-Learning-for-One-class-Classification`

Keywords: One-Class Classification, Kernel Learning, Outlier Detection, Alzheimer's disease, Magnetic resonance imaging, Breast cancer.

*Corresponding author
Email addresses: `phd1501101001@iiti.ac.in & chandangautam31@gmail.com` (Chandan Gautam), `ms1804101003@iiti.ac.in & mpratik995@gmail.com` (Pratik K Mishra), `artiwari@iiti.ac.in` (Aruna Tiwari), `phd1701241001@iiti.ac.in` (Bharat Richhariya), `pandeyh@edgehill.ac.uk` (Hari Mohan Pandey), `shuihuawang@ieee.org` (Shuihua Wang), `mtanveer@iiti.ac.in` (M. Tanveer)

[1]Data used in preparation of this article are 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`

1. Introduction

In recent years, one-class classification (OCC) has been an area of extensive research for outlier detection or anomaly detection. While conventional classification techniques aim to classify a data object into one of the many available classes, OCC aims merely to tell whether a data instance belongs to a particular class or not. A one-class classifier trains the model using samples from only one-class. This class is termed as target class. Samples, which do not belong to the target class, are treated as outliers. The one-class classifier is useful in those cases where samples from other classes are not available, or very few samples are available. Samples may not be available due to various reasons, like the difficulty of collection, high computational cost, infrequent event, etc. Coming to real-world scenarios like patient classification based on fMRI response [1], fault detection [2], document classification [3], disease detection [4], video surveillance [5, 6] where collecting data for outlier class is much more difficult and expensive, OCC is much apter than traditional multi-class classification. Further, this section is divided into three parts. The first part provides a brief survey on one-class classifiers. The second part provides a brief survey of the application of one-class classifiers in the field of biomedical data analysis. The third part gives a brief survey of deep kernel learning, an introduction of the proposed method, and its key advantages over existing one-class classifiers.

Tax [7] broadly divided one-class classifiers into three categories viz., (i) density-based classifiers (ii) boundary-based classifiers (iii) reconstruction-based classifiers. In density-based classifiers, classification is performed by estimating the density of the training data and applying a threshold on this density. It requires a large number of training samples to overcome the curse of dimensionality. This approach is very advantageous when a good probability model is assumed and the sample size is sufficient. Different density methods were applied in the past, namely, the Gaussian density method, the mixture of Gaussians, and the Parzen density [8]. Parzen density estimation [9], which is among the early works on OCC, tried to estimate the probability density of the target using the training data. It rejected the samples whose estimated probability is lower than a certain threshold. The issue with this method is that it requires a large number of training samples. The boundary-based classifiers try to obtain an optimal closed boundary around the target class. The advantage of boundary-based methods is that the number of samples needed is less in comparison to density-based methods. However, as they heavily rely on the separation between objects, they tend to be sensitive to the relative distance between the features. Boundary-based classifiers can be divided into two types viz., non-kernel and kernel-based. k-centers method [10] and k-nearest neighbors [11] are non-kernel-based classifiers. Kernel-based methods were developed by considering support vector machine (SVM) as a base classifier. Scholkopf [12] developed a boundary-based classifier that uses a hyperplane which is at a maximum distance from the origin and separates the region that contains no data. This method is referred to as one-class support vector machine (OCSVM). Tax and Duin [13] proposed another SVM-based approach where instead of a hyperplane, they used a hypersphere

to include maximum training data with minimum radius. It is referred to as support vector data description (SVDD). In reconstruction-based methods, prior knowledge of the data is used to choose a model and make assumptions about the generating process. The model is then fit to data. Here, it is assumed that the outlier objects do not satisfy the assumptions about target distribution and their reconstruction error should be high. Various one-class classifiers were developed by taking various methods as base classifiers in reconstruction-based methods. Jiang et al. [14] developed a k-means clustering-based one-class classifier. Carpenter et al. [15] developed a Learning Vector Quantization (LVQ) based one-class classifier. Bishop et al. [16] developed a Principal component analysis (PCA) based one-class classifier. Auto-Encoder or Multi-layer Perceptron (MLP) [17] and diabolo networks [18, 19] are neural network-based methods that learn to represent the input pattern at the output layer while minimizing the reconstruction error. A linear programming one-class classifier was proposed [20] that reduces the volume of the prism, imposing constraints on dissimilarity representations. Ensemble-based classifiers are another type of classifier where the main idea is to integrate multiple classifiers to obtain one that outperforms every single one of them. One such ensemble-based one-class classifier is One Class Random Forests (OCRF) [21], which combines several weak classifiers known to be accurate. It increases the generalization performance over single classifiers. It subsamples the training dataset in order to generate outliers efficiently. For a more detailed review of the OCC methods refer to the survey papers [22, 23, 24, 25, 26].

Classifier	Type	Characteristics
OCRF [21]	Ensemble-based	It combines a diverse ensemble of weak and unstable classifiers known to be accurate and increases the generalization performance over single classifiers. It subsamples the training dataset, in order to efficiently generate outliers.
Naive Parzen [27]	Density-based	The estimated density is a mixture of, most often, gaussian kernels centered on the individual training objects. It requires a large number of training samples.
k-means [14]	Reconstruction-based	It assumes that the data is clustered and can be characterized by a few prototype objects. Here, the target objects are represented by the nearest prototype vector measured by the euclidean distance.
k-NN [11]	Boundary-based	It avoids explicit density estimation and only uses distances to the first nearest neighbor. A test object is accepted when its local density is larger or equal to the local density of its (first) nearest neighbor in the training set.
Auto-Encoder (or MLP) [7]	Reconstruction-based	It is a neural network-based approach to learn a representation of the data. The difference between the input and output pattern is used as a characterization of the target class.
PCA [16]	Reconstruction-based	It describes the target data by a linear subspace. The reconstruction error is calculated to check if a new object fits the target subspace.
MST [28]	Boundary-based	A minimum spanning tree is fitted on the training data. The distance to the edges is used as a similarity metric to the target class.
k-centers [29]	Boundary-based	It covers the dataset with k small balls of equal radii. The method is sensitive to the outliers in the training set. The number of balls k and the maximum number of retries needs to be given.
MPM [30]	Boundary-based	It tries to find a linear classifier that separates the data from the origin, rejecting maximally a specific fraction of the target data.
LPDD [20]	Boundary-based	It describes the target objects which are represented in terms of distances to other objects.
OCSVM [12]	Boundary-based	It uses a hyperplane which is at a maximum distance from the origin and separates the region that contains no data.
SVDD [13]	Boundary-based	A hypersphere is used to include maximum training data with minimum radius.
OCKELM [31]	Boundary based	A non-iterative kernel-based single-layer method where training involves optimizing output weight.
VOCKELM [32]	Boundary based	A non-iterative minimum variance embedded kernel-based single-layer method where training involves optimizing output weight.
ML-OCKELM [33]	Reconstruction+ Boundary based	A non-iterative kernel-based multi-layer method where the final layer performs OCC. The layers preceding the final layer are responsible for extracting meaningful features from input data.

Table 1: Description of state-of-the-art classifiers used for comparison.

One-class classifiers are often used in the field of biomedical data analysis [34]. Early works include the use of Parzen density estimation for identification of masses in mammograms [8]. In literature [26], kernel-based one-class

classifiers (OCSVM and SVDD) show sheer dominance compared to non-kernel based one-class classifiers. OCSVM has been applied as an outlier detector for identification of disease in the past [1, 35, 36, 37, 38, 39]. OCSVM has been used for tumor segmentation from magnetic resonance imaging (MRI) [38, 39] and detection of tuberculosis [40]. In Mourao-Miranda et al. [1], OCSVM was used for detection of depression using fMRI images of the brain. OCSVM was used to detect amyloid plaques [36], which are responsible for Alzheimer's disease. Graph-based semi-supervised OCSVM [37] was used to detect abnormal lung sounds. OCSVM has been used to detect nocturnal hypermotor seizures [41]. Recently, ELM based OCC classifiers were used for drug-drug interactions discovery [42].

Apart from the good performance of SVM-based one-class classifiers, they lack in terms of computational complexity. These classifiers consume more time due to the iterative approach to learning. Leng at al. [31] addresses this issue. They developed a kernel regularized least squares (KRL)[2] based one-classifier, which follows a non-iterative approach to learning. Over the past few years, various single-layer KRL-based one-class classifiers were developed by the researchers [31, 32]. Most recently, a deep KRL-based method is developed for OCC [33, 43]. In this paper, a minimum variance-embedded deep KRL-based one-class classifier (DKRLVOC) is proposed. It makes use of the idea of minimizing the variance of samples to achieve better classification results. DKRLVOC is made of multiple KRL-based Auto-Encoders (AEs), and a final OCC layer. These AEs are responsible for better feature learning. A novel minimum variance-embedded KRL-based AE is developed which minimizes the intra-class variance, the norm of the weight, and the reconstruction error to extract meaningful features. These features are passed to the final OCC layer which classifies a sample into the target class or outlier class. The key advantages of the proposed model include,

- Less computational time due to the non-iterative approach to learning.
- Minimizing the intra-class variance between the samples to improve the generalization performance.
- Better classification accuracy by the help of representation learning. It provides a better feature representation by stacking different types of AEs in a hierarchical manner.
- More effective for small-size datasets and also in the case where obtaining data for each class is very costly or not possible.

We compare the performance of the proposed method with 16 state-of-the-art one-class classifiers based on F_1 score. Table 1 describes the state-of-the-art one-class classifiers that we have used in our paper for comparison of results against our proposed model. As 1-NN is a specific case of k-NN, we have only described k-NN method in Table 1.

[2]Leng et al. developed a kernel-based one-class classifier by taking kernel extreme learning machine (KELM) as a base classifier. KELM belongs to the family of KRL. Since KRL is a more generic name, we have used the name KRL instead of KELM in our paper for the proposed method. However, we have used the same naming convention as used in the paper for KELM-based existing methods.

The motivation behind choosing these classifiers is based upon the fact that they were frequently used as benchmark classifiers in the past [20, 21, 28, 30, 33]. The advantages and disadvantages of the methods were discussed above based on density, boundary or reconstruction. To show the applicability of our proposed method on biomedical datasets, we perform tests on 13 UCI benchmark biomedical datasets. We also present an application for the diagnosis of Alzheimer's disease (AD) based on 3-D MRI image dataset. AD is a neurodegenerative disorder which primarily affects the elderly population. According to World Alzheimer's Report-2018 [44], around 50 million people are affected by this disease worldwide. Various machine learning and deep learning-based methods [45, 46, 47, 48] are employed for the detection of AD. These methods perform multi-class classification task on AD data; however, this paper focuses on OCC task. The advantage of our one-class method is that the method can be trained on MRI images of healthy subjects only. This is helpful in real-world scenarios since the availability of healthy subjects' MRI images are very high as compared to Alzheimer's subjects' MRI images. Therefore, the Alzheimer's MRI images will be treated as outliers by one-class based methods. Generally, all deep learning methods need a huge volume of data for better performance. However, it is challenging to collect a huge volume of data for AD. Therefore, a novel deep learning method is required which can be trained on a small number of samples. Various experiments were performed in this work using volume and thickness measures of brain regions for the diagnosis of Alzheimer's disease. Moreover, to show the generalization performance of our proposed model on other medical problems, we perform the detection of breast cancer from histopathological images.

The rest of the paper is organized as follows. Section 2 discusses OCSVM, SVDD and the prerequisite KRL-based one-class classifiers. Section 3 describes our proposed method. Section 4 discusses the experimental setup and performance evaluations. Finally, we conclude our paper in section 5.

2. Preliminaries

This section briefly discusses a few state-of-the-art one-class classifiers, namely, SVM, and KRL/KELM based one-class classifiers. The SVM-based classifiers are *OCSVM* and *SVDD*, discussed in sections 2.1 and 2.2, respectively. *OCSVM* uses a hyperplane, while *SVDD* uses a hypersphere to separate the outliers. The KRL/KELM^2 is a least squares-based method. The least squares method-based one-class classifiers are *OCKELM*, *VOCKELM*, and *ML-OCKELM* and these are discussed in sections 2.3, 2.4, and 2.5, respectively.

2.1. One-class SVM: OCSVM

OCSVM was proposed by Scholkopf et al. [49] to utilize the advantages offered by SVM for OCC. Given training samples, $\{\mathbf{x_i} \mid \mathbf{x_i} \in R^d, i = 1, 2, ..., \mathcal{N}\}$, where $\mathbf{x_i}$ is a training vector, all belonging to the same target class. In OCSVM,

a hyperplane is constructed that separates all the target class sample points from the origin. The distance of this hyperplane from the origin is maximized. The model is formulated in the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\omega}, \boldsymbol{\xi}, \rho} \quad & \frac{1}{2}\boldsymbol{\omega}^T\boldsymbol{\omega} - \rho + \frac{1}{\nu\mathcal{N}}\sum_{i=1}^{\mathcal{N}}\xi_i \\ \text{s.t.} \quad & \boldsymbol{\omega}^T\boldsymbol{\phi}(\mathbf{x_i}) \geq \rho - \xi_i \quad i = 1, \ldots, \mathcal{N}, \\ & \xi_i \geq 0, \qquad i = 1, \ldots, \mathcal{N}, \end{aligned} \tag{1}$$

where, $\phi(.)$ is the mapping in the feature space, $\mathcal{N}$ is the number of training samples, $\boldsymbol{\omega}$ is the weight coefficients, ν is used to decide the fraction of target samples rejected, ρ is the bias term, and $\boldsymbol{\xi} = \{\xi_i\}$, where $i = 1, 2, \ldots, \mathcal{N}$, is the error with respect to the i^{th} sample. The dual of above equation is expressed as,

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{Q}\boldsymbol{\alpha} \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq \tfrac{1}{\nu\mathcal{N}} \quad i = 1, \ldots, \mathcal{N}, \\ & \sum_{i=1}^{\mathcal{N}}\alpha_i = 1, \end{aligned} \tag{2}$$

where, $\boldsymbol{Q_{ij}} = \boldsymbol{K}(\mathbf{x_i}, \mathbf{x_j}) = \boldsymbol{\phi}(\mathbf{x_i})^T\boldsymbol{\phi}(\mathbf{x_j})$, $\boldsymbol{K}$ is the kernel matrix, $\boldsymbol{Q_{ij}}$ is the kernel matrix generated between i^{th} and j^{th} sample, and α_i is the Lagrange multiplier. The decision function $\boldsymbol{f}(\mathbf{x})$, thus obtained from above minimization problem is as follows:

$$\begin{aligned} f(\mathbf{x}) &= sign\left(\sum_{i=1}^{\mathcal{N}}\alpha_i\boldsymbol{K}(\mathbf{x_i}, \mathbf{x}) - \rho\right) \\ &= \begin{cases} 1, & \mathbf{x} \text{ belongs to target class} \\ -1, & \mathbf{x} \text{ belongs to outlier class.} \end{cases} \end{aligned} \tag{3}$$

2.2. Support Vector Data Description: SVDD

Tax et al. [13] proposed SVDD for OCC. Here, we provide an overview of SVDD and discuss its primal, dual, and decision function formulation. Given training samples, $\{\mathbf{x_i} \mid \mathbf{x_i} \in R^d, i = 1, 2, ..., \mathcal{N}\}$, where $\mathbf{x_i}$ is a training vector, all belonging to the same target class. In SVDD, a hypersphere with no superfluous space is constructed, that consists of only target class samples. The classifier can be written as the following optimization problem [13]:

$$\begin{aligned} \min_{R,\boldsymbol{a},\boldsymbol{\xi}} \quad & R^2 + C\sum_{i=1}^{\mathcal{N}} \xi_i \\ \text{s.t.} \quad & \|\boldsymbol{\phi}(\mathbf{x_i}) - \boldsymbol{a}\|^2 \leq R^2 + \xi_i \quad i = 1,\ldots,\mathcal{N}, \\ & \xi_i \geq 0, \quad i = 1,\ldots,\mathcal{N}, \end{aligned} \tag{4}$$

where, $\phi(.)$ is a mapping to the higher dimensional feature space, $\mathcal{N}$ is the number of training samples, $\boldsymbol{a}$ is the center, and R is the radius of the hypersphere. R^2 is the distance between the center of hypersphere and any of the support vectors on the boundary. The above equation can be written as the following dual,

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{\mathcal{N}} \alpha_i Q_{ii} - \boldsymbol{\alpha}^T \boldsymbol{Q}\boldsymbol{\alpha} \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C \quad i = 1,\ldots,\mathcal{N}, \\ & \sum_{i=1}^{\mathcal{N}} \alpha_i = 1, \end{aligned} \tag{5}$$

where, $\boldsymbol{Q_{ij}} = \boldsymbol{K}(\mathbf{x_i}, \mathbf{x_j}) = \boldsymbol{\phi}(\mathbf{x_i})^T \boldsymbol{\phi}(\mathbf{x_j})$, $\boldsymbol{K}$ is the kernel matrix, and α_i is the Lagrange multiplier. A test sample can be classified as a target or an outlier based on the following decision function,

$$\begin{aligned} f(\mathbf{x}) &= sign(\|\boldsymbol{\phi}(\mathbf{x}) - \boldsymbol{a}\|^2 - R^2) \\ &= \begin{cases} -1, \ \mathbf{x} \text{ belongs to target class} \\ \ \ 1, \ \mathbf{x} \text{ belongs to outlier class.} \end{cases} \end{aligned} \tag{6}$$

2.3. OCKELM

Taking a training input $\mathbf{X} = \{\mathbf{x}_i \mid \mathbf{x}_i \in R^d, i = 1, 2, ..., \mathcal{N}\}$ and output vector $\mathbf{r} = [r, ..., r]^T \in R^{\mathcal{N}}$, where $\mathbf{x}_i$ is the input vector and r is the corresponding target label, which is a real number. $\mathcal{N}$ is the number of training samples. Algorithm 1 provides a concise presentation of the OCKELM [31] model. The training involves calculating optimum output weight, $\boldsymbol{\beta}$, by solving the following optimization problem,

$$\begin{aligned} &\min_{\boldsymbol{\beta}, e_i} \ \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + \frac{1}{2}C \sum_{i=1}^{\mathcal{N}} \|e_i\|_2^2 \\ &s.t. \ \boldsymbol{\beta}^T \boldsymbol{h}(\mathbf{x}_i) = r - e_i, \ i = 1, 2, ..., \mathcal{N}, \end{aligned} \tag{7}$$

where, e_i is the training error, and $\boldsymbol{h}(\mathbf{x}_i)$ is the non-linear feature mapping for a sample $\mathbf{x}_i$. C acts as the trade-off between minimizing the output weight norm and the training error. Solving equation (7), the output weight can be

obtained as,

$$\boldsymbol{\beta} = \mathbb{H}^T \left(\frac{1}{C}\mathbb{I} + \mathbb{H}\mathbb{H}^T \right)^{-1} \mathbf{r}, \tag{8}$$

where, $\mathbb{H} = [\boldsymbol{h}(\mathbf{x}_1), \boldsymbol{h}(\mathbf{x}_2), ..., \boldsymbol{h}(\mathbf{x}_\mathcal{N})]$, and $\mathbb{I}$ is an identity matrix. Using equation (8), the network output can be expressed as,

$$\widehat{\boldsymbol{O}} = \boldsymbol{h}(\mathbf{x})\,\boldsymbol{\beta} = \boldsymbol{h}(\mathbf{x})\,\mathbb{H}^T \left(\frac{1}{C}\mathbb{I} + \mathbb{H}\mathbb{H}^T \right)^{-1} \mathbf{r}. \tag{9}$$

By making use of Mercer's conditions, $\boldsymbol{\Omega}$ is defined as a kernel matrix in ELM as $\boldsymbol{\Omega} = \mathbb{H}\mathbb{H}^T$, where $\Omega_{j,k} = \boldsymbol{h}(\mathbf{x}_j)\boldsymbol{h}(\mathbf{x}_k) = K(\mathbf{x}_j, \mathbf{x}_k)$, $j,k = 1, ..., \mathcal{N}$ and K is a kernel function. Finally after replacing $\mathbb{H}\mathbb{H}^T$ in equation (9) with $\boldsymbol{\Omega}$, the output weight $\boldsymbol{\beta}$ is calculated as,

$$\boldsymbol{\beta} = \left(\frac{1}{C}\mathbb{I} + \boldsymbol{\Omega} \right)^{-1} \mathbf{r}. \tag{10}$$

The network output $\widehat{\boldsymbol{O}}$ of OCKELM is further calculated as,

$$\widehat{\boldsymbol{O}} = \begin{bmatrix} K(\mathbf{x}, \mathbf{x}_1) \\ \vdots \\ K(\mathbf{x}, \mathbf{x}_\mathcal{N}) \end{bmatrix}^T \left(\frac{1}{C}\mathbb{I} + \boldsymbol{\Omega} \right)^{-1} \mathbf{r}. \tag{11}$$

The distances of the network outputs to the target class is then determined as, $\boldsymbol{s} = \left| \widehat{\boldsymbol{O}} - \mathbf{r} \right|$. Larger the value of s_i, more deviant is the training sample x_i from target class. Denoting the sorted vector $\boldsymbol{s}$ as $\tilde{\boldsymbol{s}}$, the threshold (θ) is then calculated as,

$$\theta = \tilde{\boldsymbol{s}}(\lfloor \delta * \mathcal{N} \rfloor), \tag{12}$$

where, δ is the percentage of dismissal. For a t^{th} test sample $\mathbf{x}_t$, the network output $\widehat{\boldsymbol{O}}_t$ is determined as,

$$\widehat{\boldsymbol{O}}_t = \begin{bmatrix} K(\mathbf{x}_t, \mathbf{x}_1) \\ \vdots \\ K(\mathbf{x}_t, \mathbf{x}_\mathcal{N}) \end{bmatrix}^T \boldsymbol{\beta}. \tag{13}$$

The error between the network output $\widehat{\boldsymbol{O}}_t$ and target class $\mathbf{r}$ is determined as $\boldsymbol{s}_t = \left| \widehat{\boldsymbol{O}}_t - \mathbf{r} \right|$. Finally classification is done using following rule,

$$\begin{aligned} sign(\theta - \boldsymbol{s}_t) = 1, \quad & \mathbf{x}_t \textit{ belongs to target class.} \\ -1, \quad & \mathbf{x}_t \textit{ belongs to outlier class.} \end{aligned} \tag{14}$$

Algorithm 1 OCKELM

Given:

Training dataset: $\mathbf{X}$, Regularization parameter: C.

Training:

1: Calculate kernel matrix $\boldsymbol{\Omega}$ and output weight $\boldsymbol{\beta}$ using (10).
2: Calculate network output $\widehat{\boldsymbol{O}}$ using (11).
3: Calculate threshold θ using (12).

Testing:

1: For a t^{th} test sample $\mathbf{x}_t$, calculate network output $\widehat{\boldsymbol{O}}_t$ using (13).
2: Classify $\mathbf{x}_t$ using (14).

2.4. VOCKELM

Minimum Variance One-Class KELM [32] reduces the training error and intra-class variance to improve the performance of OCC. The subclasses are determined using the k-means method. Algorithm 2 provides a concise presentation of the VOCKELM model. The training model becomes minimizing the data dispersion as well as the training error using the following optimization problem,

$$\begin{aligned} &\min_{\boldsymbol{\beta}, e_i} \frac{1}{2} \left\| \boldsymbol{\beta}^T (\mathbf{V}_C + \lambda \mathbb{I}) \boldsymbol{\beta} \right\|_2^2 + \frac{C}{2} \sum_{i=1}^{\mathcal{N}} \|e_i\|_2^2 \\ &s.t.\ \boldsymbol{\beta}^T \boldsymbol{h}(\mathbf{x}_i) = r - e_i,\ i = 1, 2, ..., \mathcal{N}, \end{aligned} \tag{15}$$

where, e_i is the training error, and $\boldsymbol{h}(\mathbf{x}_i)$ is the non-linear feature mapping for a sample $\mathbf{x}_i$. $\boldsymbol{\beta}$ is the output weight, and C acts as the trade-off between the norm of output weight and the training error. r is the target class, and $\mathbb{I}$ is an identity matrix. λ is a regularization parameter adopted to avoid singularity issues with the scatter matrix $\mathbf{V}_C$, which is calculated as follows,

$$\begin{aligned} \mathbf{V}_C &= \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \left(\boldsymbol{h}(\mathbf{x}_i) - \overline{\mathbb{H}} \right) \left(\boldsymbol{h}(\mathbf{x}_i) - \overline{\mathbb{H}} \right)^T \\ &= \frac{1}{\mathcal{N}} \mathbb{H} \left(\mathbb{I} - \frac{1}{\mathcal{N}} \mathbf{u}\mathbf{u}^T \right) \mathbb{H}^T, \\ &= \mathbb{H}\, \boldsymbol{\mathcal{M}}\, \mathbb{H}^T, \end{aligned} \tag{16}$$

where, $\mathbf{u}$ is a vector of ones and $\mathbb{H} = [\boldsymbol{h}(\mathbf{x}_1),\ \boldsymbol{h}(\mathbf{x}_2),\ \ldots,\boldsymbol{h}(\mathbf{x}_\mathcal{N})]$. $\overline{\mathbb{H}} = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \boldsymbol{h}(\mathbf{x}_i)$ is the mean vector of the samples in the non-linear feature space and $\boldsymbol{\mathcal{M}}$ represents any Laplacian matrix. Replacing $\mathbf{V}_C$ in equation (15) with the

expression in equation (16) and solving the resulting optimization problem, the output weight $\boldsymbol{\beta}$ is derived as,

$$\boldsymbol{\beta} = \mathbb{H}^T \left(\mathbb{H}\mathbb{H}^T + \frac{1}{C}\boldsymbol{\mathcal{M}}\mathbb{H}\mathbb{H}^T + \frac{\lambda}{C}\mathbb{I} \right)^{-1} \mathbf{r}, \tag{17}$$

where, $\mathbf{r} = [r, ..., r]^T \in R^{\mathcal{N}}$ is a vector of target label r. The network output is expressed as $\widehat{\boldsymbol{O}} = \boldsymbol{h}(\mathbf{x})\boldsymbol{\beta}$. Applying kernelized feature mapping and defining kernel matrix $\boldsymbol{\Omega}$ in ELM as $\boldsymbol{\Omega} = \mathbb{H}\mathbb{H}^T$, where $\Omega_{j,k} = \boldsymbol{h}(\mathbf{x}_j)\boldsymbol{h}(\mathbf{x}_k) = K(\mathbf{x}_j, \mathbf{x}_k)$, $j, k = 1, ..., \mathcal{N}$ and K is a kernel function, the output weight $\boldsymbol{\beta}$ is calculated as,

$$\boldsymbol{\beta} = \left(\boldsymbol{\Omega} + \frac{1}{C}\boldsymbol{\mathcal{M}}\boldsymbol{\Omega} + \frac{\lambda}{C}\mathbb{I} \right)^{-1} \mathbf{r}. \tag{18}$$

The network output $\widehat{\boldsymbol{O}}$ is then calculated as,

$$\widehat{\boldsymbol{O}} = \begin{bmatrix} K(\mathbf{x}, \mathbf{x}_1) \\ \vdots \\ K(\mathbf{x}, \mathbf{x}_{\mathcal{N}}) \end{bmatrix}^T \left(\boldsymbol{\Omega} + \frac{1}{C}\boldsymbol{\mathcal{M}}\boldsymbol{\Omega} + \frac{\lambda}{C}\mathbb{I} \right)^{-1} \mathbf{r}, \tag{19}$$

where, $\mathbb{I}$ is an identity matrix. During training, a threshold θ is determined as $\theta = \delta\,\overline{\boldsymbol{O}}$, where $\overline{\boldsymbol{O}}$ is the mean network output of training samples and δ is the percentage of dismissal. During testing, the network output for the t^{th} test sample $\mathbf{x}_t$ is determined by,

$$\widehat{\boldsymbol{O}}_t = \begin{bmatrix} K(\mathbf{x}_t, \mathbf{x}_1) \\ \vdots \\ K(\mathbf{x}_t, \mathbf{x}_{\mathcal{N}}) \end{bmatrix}^T \boldsymbol{\beta}. \tag{20}$$

$\mathbf{x}_t$ then belongs to the target class if it satisfies the following condition,

$$\left(\widehat{\boldsymbol{O}}_t - \mathbf{r} \right)^2 \leq \theta. \tag{21}$$

2.5. ML-OCKELM

The multi-layer one-class KELM [33] employs multiple Auto-Encoders (AEs) for feature learning and a final classification layer. Algorithm 3 provides a concise presentation of the ML-OCKELM model. Assuming Q AEs are

Algorithm 2 VOCKELM

Given:

Training dataset: $\mathbf{X}$, Regularization parameter: C, Graph regularization parameter: λ

Training:

1: Calculate kernel matrix $\boldsymbol{\Omega}$ and output weight $\boldsymbol{\beta}$ using (18).
2: Calculate network output $\widehat{\boldsymbol{O}}$ using (19).
3: Calculate threshold θ as $\theta = \delta\,\overline{\boldsymbol{O}}$.

Testing:

1: For a t^{th} test sample $\mathbf{x}_t$, calculate network output $\widehat{\boldsymbol{O}}_t$ using (20).
2: Classify $\mathbf{x}_t$ using (21).

used for feature extraction, learning of features is done by optimizing the output weight $\boldsymbol{\beta}^{(q)}$ of the q^{th} AE,

$$\min_{\boldsymbol{\beta}^{(q)},\mathbf{e}_i^{(q)}} \frac{1}{2}\left\|\boldsymbol{\beta}^{(q)}\right\|_F^2 + \frac{1}{2}C^{(q)}\sum_{i=1}^{\mathcal{N}}\left\|\mathbf{e}_i^{(q)}\right\|_2^2 \tag{22}$$
$$s.t.\ \left(\boldsymbol{\beta}^{(q)}\right)^T \boldsymbol{h}\left(\mathbf{x}_i^{(q)}\right) = \mathbf{x}_i^{(q)} - \mathbf{e}_i^{(q)},\ i = 1, 2, ..., \mathcal{N},\ q = 1, 2, ..., Q,$$

where, $\boldsymbol{h}\left(\mathbf{x}_i^{(q)}\right)$ is the non-linear feature mapping and $\mathbf{e}_i^{(q)}$ is the reconstruction error for the input $\mathbf{x}_i^{(q)}$ of the q^{th} AE. $C^{(q)}$ is the regularization parameter of the q^{th} AE and $\|\,.\,\|$ refers to frobenius norm. From equation (22), the optimal $\boldsymbol{\beta}^{(q)}$ is derived as,

$$\boldsymbol{\beta}^{(q)} = \left(\mathbb{H}^{(q)}\right)^T\left(\frac{1}{C^{(q)}}\mathbb{I} + \mathbb{H}^{(q)}\left(\mathbb{H}^{(q)}\right)^T\right)^{-1}\mathbf{X}^{(q)}, \tag{23}$$

where, $\mathbf{X}^{(q)}$ denotes the input data of the q^{th} AE and $\mathbb{H}^{(q)} = \left[\boldsymbol{h}\left(\mathbf{x}_1^{(q)}\right),\ \boldsymbol{h}\left(\mathbf{x}_2^{(q)}\right),\ ...\ ,\boldsymbol{h}\left(\mathbf{x}_{\mathcal{N}}^{(q)}\right)\right]$. With the use of Mercer's conditions, kernel matrix $\boldsymbol{\Omega}^{(q)}$ is defined as $\boldsymbol{\Omega}^{(q)} = \mathbb{H}^{(q)}\left(\mathbb{H}^{(q)}\right)^T$, where $\Omega_{j,k}^{(q)} = \boldsymbol{h}(\mathbf{x}_j^{(q)})\boldsymbol{h}(\mathbf{x}_k^{(q)}) = K(\mathbf{x}_j^{(q)}, \mathbf{x}_k^{(q)}),\ j,k = 1, ..., \mathcal{N}$ and K is a kernel function. With this, the output weight for the q^{th} AE can be expressed as,

$$\boldsymbol{\beta}^{(q)} = \left(\frac{1}{C^{(q)}}\mathbb{I} + \boldsymbol{\Omega}^{(q)}\right)^{-1}\mathbf{X}^{(q)}. \tag{24}$$

The encoded feature that becomes the input of the $(q+1)^{th}$ AE for feature learning, is expressed as,

$$\mathbf{X}^{(q+1)} = \begin{bmatrix} K(\mathbf{x}^{(q)}, \mathbf{x_1}) \\ \vdots \\ K(\mathbf{x}^{(q)}, \mathbf{x}_{\mathcal{N}}) \end{bmatrix}^T \left(\frac{1}{C^{(q)}}\mathbb{I} + \boldsymbol{\Omega}^{(q)}\right)^{-1}\mathbf{X}^{(q)} \quad q = 1, ..., Q, \tag{25}$$

where, $\mathbf{X}^{(q)}$ is the input of the q^{th} AE. Also, $\mathbf{X}^{(Q+1)}$ is used to refer $\mathbf{X}^{(f)}$ which is input to the final layer. The final layer is a classification layer, with the following optimization problem,

$$\begin{aligned} &\min_{\boldsymbol{\beta}^{(f)}, e_i^{(f)}} \ \frac{1}{2} \left\|\boldsymbol{\beta}^{(f)}\right\|_2^2 + \frac{1}{2} C^{(f)} \sum_{i=1}^{\mathcal{N}} \left\|e_i^{(f)}\right\|_2^2 \\ &s.t. \ \left(\boldsymbol{\beta}^{(f)}\right)^T \boldsymbol{h}\left(\mathbf{x}_i^{(f)}\right) = r - e_i^{(f)}, \ i = 1, 2, ..., \mathcal{N}, \end{aligned} \tag{26}$$

where, $e_i^{(f)}$ is the training error and $\boldsymbol{h}\left(\mathbf{x}_i^{(f)}\right)$ is the non-linear feature mapping for a input $\mathbf{x}_i^{(f)}$ of the final layer. Solving equation (26), the output weight of the final layer is expressed as,

$$\boldsymbol{\beta}^{(f)} = \left(\frac{1}{C^{(f)}}\mathbb{I} + \boldsymbol{\Omega}^{(f)}\right)^{-1} \mathbf{r}, \tag{27}$$

where, $\mathbf{r} = [r, ..., r]^T \in R^{\mathcal{N}}$ is a vector of target label r. The network output of ML-OCKELM during training is calculated as,

$$\widehat{\boldsymbol{O}} = \begin{bmatrix} K(\mathbf{x}^{(f)}, \mathbf{x_1}) \\ \vdots \\ K(\mathbf{x}^{(f)}, \mathbf{x_{\mathcal{N}}}) \end{bmatrix}^T \left(\frac{1}{C^{(f)}}\mathbb{I} + \boldsymbol{\Omega}^{(f)}\right)^{-1} \mathbf{r}. \tag{28}$$

The distance $\boldsymbol{s}$ of the network output to the target class is then calculated as $\boldsymbol{s} = \left|\widehat{\boldsymbol{O}} - \mathbf{r}\right|$ and is sorted in decreasing order. Denoting the sorted vector $\boldsymbol{s}$ as $\tilde{\boldsymbol{s}}$, the threshold (θ) is then calculated as,

$$\theta = \tilde{\boldsymbol{s}}(\lfloor \delta * \mathcal{N} \rfloor), \tag{29}$$

where, δ is the percentage of dismissal and $\mathcal{N}$ is the number of training samples. During testing, when the t^{th} test sample $\mathbf{x}_t$ is fed to the trained model, the encoded input, $\mathbf{x}_t^{(q+1)}$, for the $(q+1)^{th}$ layer is calculated as,

$$\mathbf{x}_t^{(q+1)} = \begin{bmatrix} K(\mathbf{x}_t^{(q)}, \mathbf{x_1}) \\ \vdots \\ K(\mathbf{x}_t^{(q)}, \mathbf{x_{\mathcal{N}}}) \end{bmatrix}^T \boldsymbol{\beta}^{(q)}, \ \ q = 1, 2, ..., Q. \tag{30}$$

Here, $\mathbf{x}_t^{(Q+1)}$ is used to refer $\mathbf{x}_t^{(f)}$ which is test input to the final layer. Finally, the test network output, denoted as $\widehat{\boldsymbol{O}}_t$, is calculated as,

$$\widehat{\boldsymbol{O}}_t = \begin{bmatrix} K(\mathbf{x}_t^{(f)}, \mathbf{x_1}) \\ \vdots \\ K(\mathbf{x}_t^{(f)}, \mathbf{x_N}) \end{bmatrix}^T \boldsymbol{\beta}^{(f)}. \tag{31}$$

The error $\boldsymbol{s}_t$ is then calculated as, $\boldsymbol{s}_t = \left|\widehat{\boldsymbol{O}}_t - \mathbf{r}\right|$ and $\mathbf{x}_t$ is classified as per the rule,

$$\begin{aligned} If\ \boldsymbol{s}_t \leq \theta, &\quad \mathbf{x}_t\ belongs\ to\ target\ class. \\ otherwise, &\quad \mathbf{x}_t\ belongs\ to\ outlier\ class. \end{aligned} \tag{32}$$

Algorithm 3 ML-OCKELM

Given:

Training dataset: $\mathbf{X}^{(1)}$, Number of AE layers: Q, Regularization parameter: $\mathrm{C}^{(q)}$ for layer $q = 1, ..., Q$ and $\mathrm{C}^{(f)}$ for final layer.

Training:

1: **for** $q = 1, 2, ..., Q$ layers **do**
2: Calculate kernel matrix $\boldsymbol{\Omega}^{(q)}$ and output weight $\boldsymbol{\beta}^{(q)}$ using (24).
3: Calculate encoded input, $\mathbf{X}^{(q+1)}$, for the next $(q+1)^{th}$ layer using (25). ▷ $\mathbf{X}^{(Q+1)}$ is used to refer $\mathbf{X}^{(f)}$
4: **end for**
5: Calculate kernel matrix $\boldsymbol{\Omega}^{(f)}$ and output weight $\boldsymbol{\beta}^{(f)}$ using (27).
6: Calculate network output $\widehat{\boldsymbol{O}}$ using (28).
7: Determine the threshold θ using (29).

Testing:

1: **for** $q = 1, 2, ..., Q$ layers **do**
2: Calculate encoded test input, $\mathbf{x}_t^{(q+1)}$, using (30). ▷ $\mathbf{x}_t^{(Q+1)}$ is used to refer $\mathbf{x}_t^{(f)}$
3: **end for**
4: Calculate network output $\widehat{\boldsymbol{O}}_t$ using (31).
5: Classify $\mathbf{x}_t$ using (32).

3. The Proposed Method

This section puts forward the proposed method: minimum **v**ariance-embedded **d**eep **k**ernel **r**egularized **l**east squares for **OCC** (DKRLVOC). The architecture of the proposed method is shown in Fig.1. It is a deep architecture, which is developed by taking **k**ernel **r**egularized **l**east squares (KRL) as a base method. DKRLVOC can also be considered as a variant of any least squares-based method, like kernel extreme learning machine, least squares SVM or kernel ridge regression. We have used generic name KRL instead of these specific names and referred the

proposed method as DKRLVOC. DKRLVOC performs better than the other existing one-class classifiers pertaining to the following characteristics:

1. Non-iterative nature resulting in less computational time in comparison to its iterative counterparts.
2. Minimizing intra-class variance to achieve better separation of outliers.
3. Multi-layer architecture taking advantage of reconstruction-based and boundary-based methods.

DKRLVOC consists of mainly three types of layers viz.,

(i) minimum **v**ariance-embedded **KRL**-based **A**uto-**E**ncoder (KRLVAE).

(ii) **KRL**-based **A**uto-**E**ncoder (KRLAE).

(iii) KRL-based one-class classifier (KRLOC)

The overall architecture of the proposed method is formed by stacking above-mentioned layers. This architecture can contain any number of layers. The first layer is formed by KRLVAE. It minimizes the intra-class variance, the norm of weight, and the reconstruction error. The second layer onward is formed by stacking KRLAEs, which minimize the norm of weight and the reconstruction error. The final layer is KRLOC, which is stacked for OCC. KRLVAE and KRLAE are reconstruction-based, and the final layer, KRLOC, is boundary-based. The aim behind using KRLVAE and KRLAE is that it helps to get refined information from features even if the input is noisy. The multiple layers of AEs fine-tune the information from noisy input and extract meaningful features over subsequent layers. Here, the variance is minimized at only first layer because minimizing variance at successive layers leads to loss of pattern between the samples. This has been verified experimentally.

The training data is denoted as $\mathbf{X}^{(1)} = \{\mathbf{x}_i^{(1)}\}$, where $\mathbf{x}_i^{(1)} = [\mathrm{x}_{i1}^{(1)}, \mathrm{x}_{i2}^{(1)}, ..., \mathrm{x}_{id}^{(1)}]$, $i = 1, 2, ..., \mathcal{N}$, is the i^{th} training input of dimension d. $\mathbf{X}^{(q)} = \{\mathbf{x}_i^{(q)}\}$, where $\mathbf{x}_i^{(q)} = [\mathrm{x}_{i1}^{(q)}, \mathrm{x}_{i2}^{(q)}, ..., \mathrm{x}_{id}^{(q)}]$, $i = 1, 2, ..., \mathcal{N}$, refers to the input of the q^{th} AE. There are Q layers of stacked AEs in the proposed method denoted as $q = 1, 2, ..., Q$. The first layer, i.e., $q = 1$, is the KRLVAE layer while the subsequent layers denoted by $q = 2, ..., Q$ are KRLAE layers responsible for learning essential information from raw features. The encoded feature output of the q^{th} AE acts as input to the $(q + 1)^{th}$ AE.

Parameter	Description	Range of values taken for experiments
Q	Number of stacked AE layers.	2
$\mathrm{C}^{(q)}$	Regularization parameter for layer $q = 1, ..., Q$.	$\{2^{-5}, 2^{-4}, ..., 2^{5}\}$
$\mathrm{C}^{(f)}$	Regularization parameter for final layer.	$\{2^{-5}, 2^{-4}, ..., 2^{5}\}$
λ	Graph regularization parameter.	1
k	Number of clusters for k-means clustering to group data into subclasses.	$\{1, 2, ..., 10\}$
δ	Percentage of dismissal of outliers.	$\{1\%, 5\%, 10\%\}$

Table 2: Model parameter descriptions.

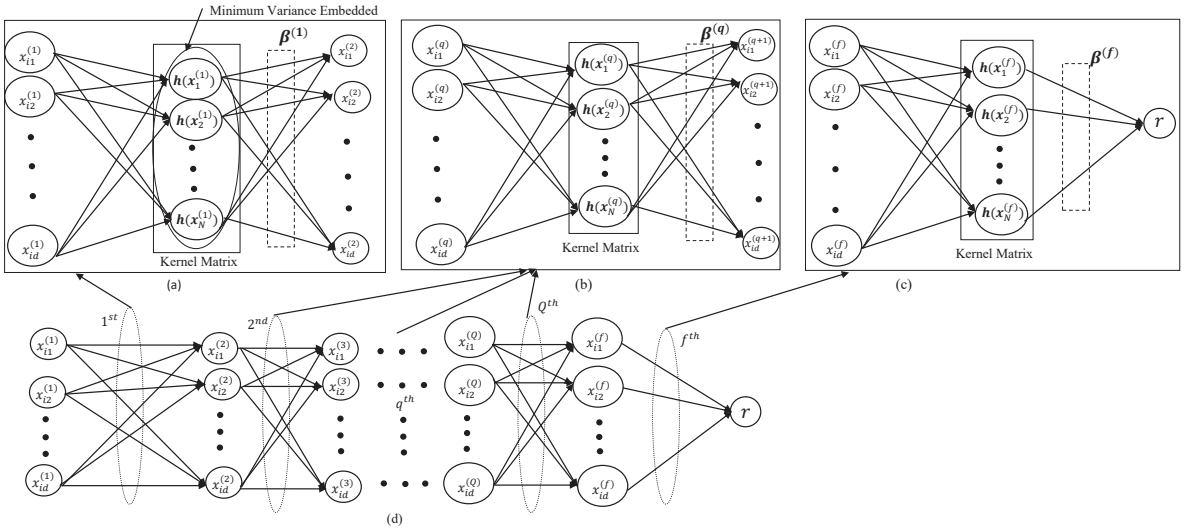


Figure 1: Architecture of DKRLVOC. (a) Encoded output of KRLVAE layer is fed as input to next KRLAE layer. (b) Encoded output of each KRLAE is fed as input to the subsequent KRLAE layer. (c) KRLOC layer takes encoded output of Q^{th} KRLAE layer as input. (d) Shows arrangement of different layers.

Finally, the encoded feature output of the Q^{th} AE acts as input to the final KRLOC layer. Table 2 provides a tabular description of the model parameters and the range of values they are selected from in the experiments. The parameter estimation process is explained in section 4. Algorithm 4 provides detailed implementation steps for the proposed model.

In the proposed DKRLVOC, the variance of the output for the first layer is calculated as,

$$\begin{aligned}
\mathbf{V}_{\omega} &= \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \left(\mathrm{X}_i^{(2)} - \overline{\mathrm{X}^{(2)}}\right) \left(\mathrm{X}_i^{(2)} - \overline{\mathrm{X}^{(2)}}\right)^T, \\
&= \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \left(\left(\boldsymbol{\beta}^{(1)}\right)^T \boldsymbol{h}(\mathbf{x}_i^{(1)}) - \left(\boldsymbol{\beta}^{(1)}\right)^T \overline{\mathbb{H}^{(1)}}\right) \left(\left(\boldsymbol{\beta}^{(1)}\right)^T \boldsymbol{h}(\mathbf{x}_i^{(1)}) - \left(\boldsymbol{\beta}^{(1)}\right)^T \overline{\mathbb{H}^{(1)}}\right)^T, \\
&= \left(\boldsymbol{\beta}^{(1)}\right)^T \left(\frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \left(\boldsymbol{h}(\mathbf{x}_i^{(1)}) - \overline{\mathbb{H}^{(1)}}\right) \left(\boldsymbol{h}(\mathbf{x}_i^{(1)}) - \overline{\mathbb{H}^{(1)}}\right)^T\right) \boldsymbol{\beta}^{(1)}, \\
&= \left(\boldsymbol{\beta}^{(1)}\right)^T \mathbf{V}_C \, \boldsymbol{\beta}^{(1)},
\end{aligned} \tag{33}$$

where, $\boldsymbol{\beta}^{(1)}$ is the output weight for the first layer, and $\boldsymbol{h}\left(\mathbf{x}_i^{(1)}\right)$ is the non-linear feature mapping for training sample $\mathbf{x}_i^{(1)}$. $\mathrm{X}_i^{(2)}$ is the encoded feature output of first layer or input to second layer for a training sample $\mathbf{x}_i^{(1)}$, $\overline{\mathrm{X}^{(2)}} = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \mathrm{X}_i^{(2)}$ is the mean output for all training samples for first layer, $\overline{\mathbb{H}^{(1)}} = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \boldsymbol{h}(\mathbf{x}_i^{(1)})$ is the mean vector of the samples in

the non-linear feature space of the first layer, and $\mathbf{V}_C$ is the scatter matrix of the training class.

Minimum variance-embedding is done at first layer, KRLVAE, by minimizing either class or intra-class variance which is encoded by the scatter matrix represented by $\mathbf{V}_C$ or $\mathbf{V}_S$, respectively. The class variance of the training data is defined as follows,

$$\mathbf{V}_C = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} (\boldsymbol{h}(\mathbf{x}_i^{(1)}) - \overline{\mathbb{H}^{(1)}})(\boldsymbol{h}(\mathbf{x}_i^{(1)}) - \overline{\mathbb{H}^{(1)}})^T, \tag{34}$$

where, $\mathbb{H}^{(1)} = \left[\boldsymbol{h}\left(\mathbf{x}_1^{(1)}\right), \boldsymbol{h}\left(\mathbf{x}_2^{(1)}\right), \ldots, \boldsymbol{h}\left(\mathbf{x}_{\mathcal{N}}^{(1)}\right)\right]$. The scatter matrix $\mathbf{V}_C$ can further be expressed as,

$$\begin{aligned}
\mathbf{V}_C &= \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} (\boldsymbol{h}(\mathbf{x}_i^{(1)}) - \overline{\mathbb{H}^{(1)}})(\boldsymbol{h}(\mathbf{x}_i^{(1)}) - \overline{\mathbb{H}^{(1)}})^T \\
&= \frac{1}{\mathcal{N}} \mathbb{H}^{(1)} \left(\mathbb{I} - \frac{1}{\mathcal{N}} \mathbf{u}\mathbf{u}^T \right) \left(\mathbb{H}^{(1)} \right)^T \\
&= \mathbb{H}^{(1)} \boldsymbol{\mathcal{M}} \left(\mathbb{H}^{(1)} \right)^T,
\end{aligned} \tag{35}$$

where, $\mathbb{I}$ is an identity matrix, $\mathbf{u}$ is a vector of ones, and $\boldsymbol{\mathcal{M}}$ represents any Laplacian matrix. The intra-class variance can be calculated as,

$$\mathbf{V}_S = \sum_{i=1}^{\mathcal{N}} \sum_{p=1}^{P} \frac{\mathcal{N}_p}{\mathcal{N}} \gamma_i^p (\boldsymbol{h}(\mathbf{x}_i^{(1)}) - \overline{\mathbb{H}^{(1)}})(\boldsymbol{h}(\mathbf{x}_i^{(1)}) - \overline{\mathbb{H}^{(1)}})^T, \tag{36}$$

where, the number of training samples belonging to a cluster p is denoted by $\mathcal{N}_p$ and γ_i^p denotes whether the sample $\mathbf{x}_i^{(1)}$ belongs to cluster p or not. A clustering method like k-means is used to group data into subclasses. $\mathbf{V}_S$ can further be expressed similarly as $\mathbf{V}_C$ in equation (35). KRLVAE can be expressed in the form of following optimization problem,

$$\begin{aligned}
&\min_{\boldsymbol{\beta}^{(1)}, \mathbf{e}_i^{(1)}} \ \frac{1}{2} Tr\left(\left(\boldsymbol{\beta}^{(1)}\right)^T \left(\mathbf{V}_C + \lambda \mathbb{I}\right) \boldsymbol{\beta}^{(1)} \right) + \frac{C^{(1)}}{2} \sum_{i=1}^{\mathcal{N}} \left\| \mathbf{e}_i^{(1)} \right\|_2^2 \\
&s.t. \ \left(\boldsymbol{\beta}^{(1)}\right)^T \boldsymbol{h}\left(\mathbf{x}_i^{(1)}\right) = \mathbf{x}_i^{(1)} - \mathbf{e}_i^{(1)}, \ i = 1, 2, ..., \mathcal{N},
\end{aligned} \tag{37}$$

where, $\mathbf{e}_i^{(1)}$ is the reconstruction error. $C^{(1)}$ acts as a trade-off between minimizing the output weight norm and the reconstruction error for the first layer. λ is the graph regularization parameter. The Langrangian relaxation of equation (37) after substituting equation (35) in equation (37) can be found as,

$$\mathcal{L}_{KRLVAE} = \frac{1}{2} Tr\left(\left(\boldsymbol{\beta}^{(1)}\right)^T \left(\mathbb{H}^{(1)} \boldsymbol{\mathcal{M}} \left(\mathbb{H}^{(1)} \right)^T + \lambda \mathbb{I} \right) \boldsymbol{\beta}^{(1)} \right) + \frac{C^{(1)}}{2} \sum_{i=1}^{\mathcal{N}} \left\| \mathbf{e}_i^{(1)} \right\|_2^2 - \sum_{i=1}^{\mathcal{N}} \alpha_i^{(1)} \left(\left(\boldsymbol{\beta}^{(1)}\right)^T \boldsymbol{h}\left(\mathbf{x}_i^{(1)}\right) - \mathbf{x}_i^{(1)} + \mathbf{e}_i^{(1)} \right), \tag{38}$$

where, $\alpha^{(1)} = \{\alpha_i^{(1)}\}$, $i = 1, 2, ..., \mathcal{N}$, is a Langrangian multiplier of first layer. We optimize equation (38) by computing

its derivatives as follows:

$$\frac{\partial \mathcal{L}_{KRLVAE}}{\partial \boldsymbol{\beta}^{(1)}} = 0 \implies \boldsymbol{\beta}^{(1)} = \alpha^{(1)} \mathbb{H}^{(1)} \left(\mathbb{H}^{(1)} \boldsymbol{\mathcal{M}} \left(\mathbb{H}^{(1)} \right)^T + \lambda \mathbb{I} \right)^{-1}, \tag{39}$$

$$\frac{\partial \mathcal{L}_{KRLVAE}}{\partial \mathbf{e}_i^{(1)}} = 0 \implies \mathbf{E}^{(1)} = \frac{\alpha^{(1)}}{C^{(1)}}, \tag{40}$$

$$\frac{\partial \mathcal{L}_{KRLVAE}}{\partial \alpha^{(1)}} = 0 \implies \alpha^{(1)} = C^{(1)} \left(\mathbf{X}^{(1)} - \left(\boldsymbol{\beta}^{(1)} \right)^T \mathbb{H}^{(1)} \right). \tag{41}$$

Substituting equation (41) in equation (39), we get,

$$\boldsymbol{\beta}^{(1)} = \mathbb{H}^{(1)} \left(\mathbb{H}^{(1)} \left(\mathbb{H}^{(1)} \right)^T + \frac{\mathbb{H}^{(1)} \boldsymbol{\mathcal{M}} \left(\mathbb{H}^{(1)} \right)^T}{C^{(1)}} + \frac{\lambda}{C^{(1)}} \mathbb{I} \right)^{-1} \mathbf{X}^{(1)}. \tag{42}$$

Applying kernelized feature mapping by defining $\boldsymbol{\Omega}^{(1)} = \mathbb{H}^{(1)} \left(\mathbb{H}^{(1)} \right)^T$, where $\Omega^{(1)}_{j,k} = \boldsymbol{h} \left(\mathbf{x}_j^{(1)} \right) \boldsymbol{h} \left(\mathbf{x}_k^{(1)} \right) = K \left(\mathbf{x}_j^{(1)}, \mathbf{x}_k^{(1)} \right)$, $j, k = 1, ..., \mathcal{N}$ and K is a kernel function, we can re-write equation (42) as,

$$\boldsymbol{\beta}^{(1)} = \left(\boldsymbol{\Omega}^{(1)} + \frac{\boldsymbol{\mathcal{M}} \boldsymbol{\Omega}^{(1)}}{C^{(1)}} + \frac{\lambda}{C^{(1)}} \mathbb{I} \right)^{-1} \mathbf{X}^{(1)}. \tag{43}$$

The encoded feature that becomes the input of the KRLAE layer is then calculated as,

$$\mathbf{X}^{(2)} = \begin{bmatrix} K(\mathbf{x}^{(1)}, \mathbf{x_1}) \\ \vdots \\ K(\mathbf{x}^{(1)}, \mathbf{x_{\mathcal{N}}}) \end{bmatrix}^T \left(\boldsymbol{\Omega}^{(1)} + \frac{\boldsymbol{\mathcal{M}} \boldsymbol{\Omega}^{(1)}}{C^{(1)}} + \frac{\lambda}{C^{(1)}} \mathbb{I} \right)^{-1} \mathbf{X}^{(1)}. \tag{44}$$

Thereafter $(Q-1)$ KRLAE layers are used to extract meaningful features, where the encoded output of one KRLAE layer becomes the input of the next KRLAE layer. The output weight in the KRLAE layers is optimized,

$$\min_{\boldsymbol{\beta}^{(q)}, \mathbf{e}_i^{(q)}} \frac{1}{2} \left\| \boldsymbol{\beta}^{(q)} \right\|_F^2 + \frac{C^{(q)}}{2} \sum_{i=1}^{\mathcal{N}} \left\| \mathbf{e}_i^{(q)} \right\|_2^2 \tag{45}$$

$$s.t. \ \left(\boldsymbol{\beta}^{(q)} \right)^T \boldsymbol{h} \left(\mathbf{x}_i^{(q)} \right) = \mathbf{x}_i^{(q)} - \mathbf{e}_i^{(q)}, \ i = 1, 2, ..., \mathcal{N}, \ q = 2, 3, ..., Q,$$

where, $C^{(q)}$ acts as a trade-off between minimizing the reconstruction error and the output weight norm. $\mathbf{e}_i^{(q)}$ is the reconstruction error vector of i^{th} input of q^{th} layer. $\| \, . \, \|$ refers to frobenius norm. The Langrangian relaxation of

equation (45) can be found as,

$$\mathcal{L}_{KRLAE} = \frac{1}{2}\left\|\boldsymbol{\beta}^{(q)}\right\|_F^2 + \frac{C^{(q)}}{2}\sum_{i=1}^{\mathcal{N}}\left\|\mathbf{e}_i^{(q)}\right\|_2^2 - \sum_{i=1}^{\mathcal{N}}\alpha_i^{(q)}\left(\left(\boldsymbol{\beta}^{(q)}\right)^T \boldsymbol{h}\left(\mathbf{x}_i^{(q)}\right) - \mathbf{x}_i^{(q)} + \mathbf{e}_i^{(q)}\right), \tag{46}$$

where, $\alpha^{(q)} = \{\alpha_i^{(q)}\},\ i = 1, 2, ..., \mathcal{N}$, is the Langrangian multiplier of q^{th} layer. Equation (46) is optimized as follows:

$$\frac{\partial \mathcal{L}_{KRLAE}}{\partial \boldsymbol{\beta}^{(q)}} = 0 \implies \boldsymbol{\beta}^{(q)} = \alpha^{(q)}\mathbb{H}^{(q)}, \tag{47}$$

$$\frac{\partial \mathcal{L}_{KRLAE}}{\partial \mathbf{e}_i^{(q)}} = 0 \implies \mathbf{E}^{(q)} = \frac{\alpha^{(q)}}{C^{(q)}}, \tag{48}$$

$$\frac{\partial \mathcal{L}_{KRLAE}}{\partial \alpha^{(q)}} = 0 \implies \alpha^{(q)} = C^{(q)}\left(\mathbf{X}^{(q)} - \left(\boldsymbol{\beta}^{(q)}\right)^T \mathbb{H}^{(q)}\right). \tag{49}$$

Substituting equation (49) in equation (47), we get,

$$\boldsymbol{\beta}^{(q)} = \mathbb{H}^{(q)}\left(\frac{1}{C^{(q)}}\mathbb{I} + \mathbb{H}^{(q)}\left(\mathbb{H}^{(q)}\right)^T\right)^{-1}\mathbf{X}^{(q)}. \tag{50}$$

Substituting $\boldsymbol{\Omega}^{(q)} = \mathbb{H}^{(q)}\left(\mathbb{H}^{(q)}\right)^T$, where $\Omega_{j,k}^{(q)} = \boldsymbol{h}\left(\mathbf{x}_j^{(q)}\right)\boldsymbol{h}\left(\mathbf{x}_k^{(q)}\right) = K\left(\mathbf{x}_j^{(q)}, \mathbf{x}_k^{(q)}\right),\ j, k = 1, ..., \mathcal{N}$ and K is a kernel function, we can re-write equation (50) as,

$$\boldsymbol{\beta}^{(q)} = \left(\frac{1}{C^{(q)}}\mathbb{I} + \boldsymbol{\Omega}^{(q)}\right)^{-1}\mathbf{X}^{(q)}. \tag{51}$$

The encoded feature that becomes the input of the $(q+1)^{th}$ AE, is expressed as,

$$\mathbf{X}^{(q+1)} = \begin{bmatrix} K(\mathbf{x}^{(q)}, \mathbf{x_1}) \\ \vdots \\ K(\mathbf{x}^{(q)}, \mathbf{x}_{\mathcal{N}}) \end{bmatrix}^T \left(\frac{1}{C^{(q)}}\mathbb{I} + \boldsymbol{\Omega}^{(q)}\right)^{-1}\mathbf{X}^{(q)},\ \ q = 2, 3, ..., Q. \tag{52}$$

Here, $\mathbf{X}^{(Q+1)}$ is used to refer $\mathbf{X}^{(f)}$ which is input to the final layer. At the final KRLOC layer, the output weight $\boldsymbol{\beta}^{(f)}$ is derived using the following optimization problem,

$$\min_{\boldsymbol{\beta}^{(f)}, e_i^{(f)}} \frac{1}{2}\left\|\boldsymbol{\beta}^{(f)}\right\|_2^2 + \frac{C^{(f)}}{2}\sum_{i=1}^{\mathcal{N}}\left\|e_i^{(f)}\right\|_2^2 \tag{53}$$

$$s.t.\ \left(\boldsymbol{\beta}^{(f)}\right)^T \boldsymbol{h}\left(\mathbf{x}_i^{(f)}\right) = r - e_i^{(f)},\ i = 1, 2, ..., \mathcal{N},$$

where, $e_i^{(f)}$ is the training error and $\boldsymbol{h}\left(\mathbf{x}_i^{(f)}\right)$ is the non-linear feature mapping for input $\mathbf{x}_i^{(f)}$ of final layer. r is the target class. Solving the above minimization problem in a similar fashion as equation (45), the final output weight $\boldsymbol{\beta}^{(f)}$ is derived as,

$$\boldsymbol{\beta}^{(f)} = \left(\frac{1}{C^{(f)}}\mathbb{I} + \boldsymbol{\Omega}^{(f)}\right)^{-1} \mathbf{r}, \tag{54}$$

where, $\mathbf{r} = [r, ..., r]^T \in R^{\mathcal{N}}$ is target class vector. The network output of the proposed method during training, denoted as $\widehat{\boldsymbol{O}}$, can then be calculated as,

$$\widehat{\boldsymbol{O}} = \begin{bmatrix} K(\mathbf{x}^{(f)}, \mathbf{x_1}) \\ \vdots \\ K(\mathbf{x}^{(f)}, \mathbf{x_{\mathcal{N}}}) \end{bmatrix}^T \left(\frac{1}{C^{(f)}}\mathbb{I} + \boldsymbol{\Omega}^{(f)}\right)^{-1} \mathbf{r}. \tag{55}$$

The threshold (θ) is further determined during training as follows:

(i) For each training sample $\mathbf{x}_i$, the distance between network output $\widehat{O}_i$ and target label r is calculated as,

$$s(i) = \left|\widehat{O}_i - r\right|. \tag{56}$$

(ii) The vector $\boldsymbol{s}$ is sorted in decreasing order, denoted as, $\tilde{\boldsymbol{s}}$. A small portion of training data is dismissed as outliers starting from the most deviant ones as they are probably the most distant from the target class distribution. The threshold is then decided as,

$$\theta = \tilde{\boldsymbol{s}}\left(\lfloor \delta * \mathcal{N} \rfloor\right), \; 0 < \delta \leq 1, \tag{57}$$

where, δ is the percentage of dismissal and $\mathcal{N}$ is the amount of training data. $\lfloor . \rfloor$ refers to floor value.

During testing, when the t^{th} test sample $\mathbf{x}_t$ is fed to the trained model, the encoded input, $\mathbf{x}_t^{(q+1)}$, for $(q+1)^{th}$ layer is calculated as,

$$\mathbf{x}_t^{(q+1)} = \begin{bmatrix} K(\mathbf{x}_t^{(q)}, \mathbf{x_1}) \\ \vdots \\ K(\mathbf{x}_t^{(q)}, \mathbf{x_{\mathcal{N}}}) \end{bmatrix}^T \boldsymbol{\beta}^{(q)}, \;\; q = 1, 2, ..., Q. \tag{58}$$

Here, $\mathbf{x}_t^{(Q+1)}$ is used to refer $\mathbf{x}_t^{(f)}$ which is test input to the final layer. Finally, the test network output, denoted as $\widehat{\boldsymbol{O}}_t$, is calculated as,

$$\widehat{\boldsymbol{O}}_t = \begin{bmatrix} K(\mathbf{x}_t^{(f)}, \mathbf{x_1}) \\ \vdots \\ K(\mathbf{x}_t^{(f)}, \mathbf{x_{\mathcal{N}}}) \end{bmatrix}^T \boldsymbol{\beta}^{(f)}. \tag{59}$$

Algorithm 4 DKRLVOC

Given:

Training dataset: $\mathbf{X}^{(1)}$, Number of AE layers: Q, Regularization parameter: $\mathrm{C}^{(q)}$ for layer $q = 1, ..., Q$ and $\mathrm{C}^{(f)}$ for final layer, Graph regularization parameter: λ

Training:

for $q = 1, 2, ..., Q$ layers **do**
  if $q == 1$ **then**
    Calculate kernel matrix $\boldsymbol{\Omega}^{(\mathbf{1})}$ and output weight $\boldsymbol{\beta}^{(\mathbf{1})}$ using (43) for the first layer (,i.e., KRLVAE).
    Calculate encoded input, $\mathbf{X}^{(2)}$, for the second layer using (44).
  else
    Calculate kernel matrix $\boldsymbol{\Omega}^{(\boldsymbol{q})}$ and output weight $\boldsymbol{\beta}^{(\boldsymbol{q})}$ using (51) for KRLAE at q^{th} layer.
    Calculate encoded input, $\mathbf{X}^{(q+1)}$, for the subsequent KRLAE layer using (52). ▷ $\mathbf{X}^{(Q+1)}$ is used to refer $\mathbf{X}^{(f)}$
  end if
end for
Calculate kernel matrix $\boldsymbol{\Omega}^{(f)}$ and output weight $\boldsymbol{\beta}^{(f)}$ using (54) for the final layer (,i.e., KRLOC).
Calculate network output $\widehat{\boldsymbol{O}}$ using (55).
Finally, calculate threshold θ using (57).

Testing:

for $q = 1, 2, ..., Q$ layers **do**
  Calculate encoded test input, $\mathbf{x}_t^{(q+1)}$, for the subsequent KRLAE layer using (58). ▷ $\mathbf{x}_t^{(Q+1)}$ is used to refer $\mathbf{x}_t^{(f)}$
end for
Calculate network output $\widehat{\boldsymbol{O}}_t$ using (59).
Calculate distance s_t and classify $\mathbf{x}_t$ using (60).

The distance s_t for test data is then calculated as, $s_t = \left|\widehat{\boldsymbol{O}}_t - \mathbf{r}\right|$. Finally, the decision is made based on the following rule,

$$If\ s_t \leq \theta, then\ \mathbf{x}_t\ belongs\ to\ target\ class. \tag{60}$$
$$Otherwise, \mathbf{x}_t\ belongs\ to\ outlier\ class.$$

4. Experiments

We conduct experiments on 18 benchmark (13 biomedical and 5 non-biomedical) datasets and 2 real-world biomedical datasets. Further, these 13 biomedical benchmark datasets can be categorized as 11 small-size and 2 medium-size UCI benchmark datasets. These 5 non-biomedical datasets can be categorized as 3 small-size and 2 medium-size UCI benchmark datasets. To show the applicability of DKRLVOC on real-world biomedical datasets, we utilize image data for the diagnosis of Alzheimer's and Breast Cancer disease. We convert all binary or multi-class datasets into one-class datasets. We do this conversion by taking one of the classes as target class and the rest of the classes as outliers [7]. The target class for these datasets is mentioned in Table 3, 7, 8 and 9.

Matlab R2016a is used for all the trials running on a PC with Intel Core i5 3.10GHz CPU, 32 GB RAM. We perform experiments for the proposed method, DKRLVOC, by taking the number of stacked AE layers (Q) as 2 and a final OCC layer. Minimum variance embedding is done at the first layer. The graph regularization parameter λ at first layer is taken as 1. 5-fold cross-validation is used to select the optimal parameters during the time of training from the range of values provided in Table 2. All the methods employ the Radial Basis Function (RBF) kernel, which can be calculated for data points $\mathbf{x_i}$ and $\mathbf{x_j}$ as follows:

$$\boldsymbol{k}(\mathbf{x_i}, \mathbf{x_j}) = exp\left(-\frac{\left\|\mathbf{x_i} - \mathbf{x_j}\right\|_2^2}{2\sigma^2}\right), \tag{61}$$

where, σ is derived by determining the mean of the euclidean distance across different training samples. The experimental setup of all the methods is kept the same to facilitate a fair comparison for all the datasets.

Further, we compute the following measures for performance analysis:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{62}$$

$$Precision\ (P) = \frac{TP}{TP + FP}, \tag{63}$$

$$Recall\ (R) = \frac{TP}{TP + FN}, \tag{64}$$

$$Specificity = \frac{TN}{TN + FP}, \tag{65}$$

$$F_1\ score = \frac{2\ P.R}{P + R}, \tag{66}$$

$$G - mean = \sqrt{P.R}. \tag{67}$$

Above *FN*, *FP*, *TN*, and *TP* represent false negative, false positive, true negative, and true positive, respectively. Accuracy denotes the portion of all correct predictions. Precision reflects the portion of correct positive predictions among all the predicted positives. Recall indicates the portion of correct positive predictions among the actual positives, while specificity indicates the portion of correct negative predictions among the actual negatives. F_1 score is the harmonic mean of precision and recall. G-mean is the geometric mean of precision and recall. In case of imbalanced data, if the model is doing good towards classifying samples of majority class, and poor towards the class with fewer samples, the accuracy may still give an impression that the performance is overall good, simply because the model may be classifying most of the samples to majority class. So accuracy fails to give a complete picture of the perfor-

mance of a model. This inability of the accuracy to give a proper estimate is further explained in greater detail in section 4.2. Precision and recall give a better idea of the performance of a model. However, taking only one of either precision or recall is not the best performance metric for some applications. In such cases, F_1 score and g-mean are used to seek a balance between precision and recall. Hence, either of them can be used as a comparison metric in case of imbalanced datasets. In this paper, we have primarily used F_1 score as the comparison metric.

We compare the performance of the proposed method, DKRLVOC, with 16 existing one-class methods, namely, One Class Random Forests (OCRF) [21], Naive Parzen density estimation [27], k-means [14], 1-Nearest Neighbor (1-NN) [50], k-Nearest Neighbor (k-NN) [11], Auto-Encoder neural network or Multi-layer Perceptron (MLP) [7], Principal Component Analysis (PCA) [16], a Minimum Spanning Tree based one-class classifier (MST) [28], k-centers [29], Minimax Probability Machine (MPM) [30], Linear Programming Dissimilarity Data Description (LPDD) [20], Support Vector Data Description (SVDD) [13], One Class Support Vector Machine (OCSVM) [12], OCKELM [31], VOCKELM [32], and ML-OCKELM [33]. The implementations of the methods are taken from ddtools [51]. OCSVM is implemented using the LIBSVM library [52].

Further, we divide this section into three parts. In the section 4.1, we discuss experimental results on small-size UCI benchmark datasets. In the section 4.2, we discuss experimental results on medium-size UCI benchmark datasets. In the section 4.3, we discuss experimental results on real-world datasets.

4.1. Experiments on small-size UCI datasets

We conduct experiments on 14 small-size UCI benchmark classification datasets (11 biomedical and 3 others). We present the details of the datasets in Table 3. The small-size datasets are the ones that contain a low amount of training data. All features are normalized using z-score with a mean 0 and standard deviation 1. The target and outlier class samples are divided into two halves each. One half of the target class sample along with one half of the outlier class sample is used for 5-fold cross-validation. The remaining half is used as the test set. It is to be noted that only samples from the target class are used to train the model. The samples with missing feature values are removed.

Table 5 provides the optimal set of parameters for small-size datasets selected for DKRLVOC using 5-fold cross-validation during training time.

This section is divided into two parts; the section 4.1.1 discusses experimental results on the biomedical small-size datasets, while the section 4.1.2 discusses the results on other small-size datasets.

4.1.1. Experiments on biomedical small-size datasets

We present the F_1 scores for different OCC methods on 11 small-size biomedical datasets in Table 4 along with the average scores for each method over all the datasets. The best results are highlighted in bold. The proposed method,

S.no.	Datasets	#Total Samples	#Target	#Outlier	#Features	Target Class
	Biomedical datasets					
1	Arrhythmia	420	183	237	278	Abnormal
2	Biomed	194	67	127	5	Diseased
3	Breast Cancer[1]	699	458	241	9	Benign
4	Caesarian	80	34	46	5	0
5	Cancer[2]	198	151	47	33	Non Recurring
6	Cardiotocography	2126	176	1950	22	Pathologic
7	Colposcopy[3]	97	82	15	62	Good
8	Cryotherapy	90	48	42	6	1
9	Hepatitis	155	123	32	19	Normal
10	SPECT Heart	349	254	95	44	Abnormal
11	Survival	306	225	81	3	Greater than 5 year
	Other datasets					
12	Glass Building	214	76	138	9	Non float
13	Ionosphere	351	126	225	34	Bad
14	Iris	150	50	100	4	Setosa

[1] Refers to Wisconsin Breast Cancer UCI dataset.
[2] Refers to Wisconsin Prognostic Breast Cancer UCI dataset.
[3] Colposcopy dataset with modality hinselmann is used for experimental purpose.

Table 3: UCI small-size dataset specifications.

	Biomedical Datasets											Other Datasets			Average
	Arrhy thmia	Biomed	Breast Cancer	Caesarian	Cancer	Cardioto cography	Colpo scopy	Cryoth erapy	Hepatitis	SPECT Heart	Survival	Glass Building	Ionos phere	Iris	
OCRF [21]	**60.67**	51.16	79.24	59.65	82.42	15.29	87.06	69.57	88.41	**84.39**	85.17	52.41	56.11	50	65.83
Naive Parzen [27]	**60.67**	45.53	90.95	59.46	73.83	39.07	81.08	77.55	86.18	79.58	83.33	51.49	62.34	78.05	69.22
k-means [14]	**60.67**	50	94.98	53.66	86.39	25.81	89.66	69.39	88.41	**84.39**	84.94	53.1	51.69	93.62	70.48
1-NN [50]	58.98	48.82	92.99	51.16	82.58	39.39	89.66	73.68	88.41	79.17	83.74	55.36	51.69	91.3	70.50
k-NN [11]	**60.67**	50	95.69	51.16	86.39	34.08	90.91	71.64	88.41	**84.39**	83.87	52.34	52.32	95.83	71.26
Auto-Encoder (or MLP) [7]	58.5	38.26	95.48	53.66	79.49	55.51	88.37	71.11	88.89	**84.39**	83.79	58.41	52.94	93.62	71.60
PCA [16]	57.44	48.54	93.51	60.38	79.75	35.58	86.75	77.78	88.89	80.41	83.27	55.93	41.62	80.95	69.34
MST [28]	**60.67**	50	95.69	51.16	86.39	34.08	90.91	**80**	88.41	**84.39**	82.95	52.34	52.32	97.96	71.95
k-centers [29]	59.73	44.04	95.28	50	84.34	26.76	89.66	74.07	88.41	84	81.36	55.1	52.32	88.89	69.57
MPM [30]	NaN	NaN	92.41	51.61	NaN	2.25	NaN	NaN	NaN	21.13	76.42	61.22	33.16	88.89	30.51
LPDD [20]	NaN	NaN	92.77	54.55	NaN	2.25	NaN	NaN	NaN	21.13	68.82	54.72	48.7	88.89	30.85
Kernel based methods															
OCSVM [12]	57.04	47.79	93.85	62.86	84.66	17.17	86.75	75.47	84.85	81.45	79.01	58.59	46.7	68.42	67.47
SVDD [13]	56.12	48.21	93.74	55.17	83.02	19.15	82.05	75.47	86.57	77.27	80.17	59.18	42.53	64.86	65.97
Single-layer methods															
OCKELM [31]	59.46	47.37	95.28	53.66	84.02	63.76	90.48	77.78	88.89	83.61	83.72	56.25	52.94	83.72	72.92
VOCKELM [32]	58.95	53.06	76.68	61.11	86.05	63.45	85	75.56	88.89	84	81.57	58.33	53.45	93.62	72.84
Multi-layer methods															
ML-OCKELM [33]	**60.67**	47.71	95.26	59.65	84.02	69.79	89.66	78.43	87.22	84	84.25	54.55	60.42	**100**	75.40
DKRLVOC	**60.67**	**53.45**	**95.96**	**62.96**	**86.71**	**70.53**	**92.13**	**80**	**89.39**	**84.39**	**85.49**	**62.5**	**66.29**	**100**	**77.89**

Table 4: F_1 score comparisons on small-size datasets.

DKRLVOC, obtains the highest F_1 score on all 11 out of 11 biomedical datasets as compared to the other OCC methods. DKRLVOC scores 0.1% $\sim$ 6.77% higher than the single-layer based methods and 0.39% $\sim$ 5.74% higher than the multi-layer based method, ML-OCKELM. DKRLVOC achieves this by reducing the variance of different subclasses formed due to the uneven distribution of data within the class. Fig.5 shows the recall, precision, g-mean and accuracy comparisons between DKRLVOC and the existing KELM methods. It can be observed that DKRLVOC generally achieves comparatively better performance than the other methods. It is quite clear from Fig.5a, that for 8 datasets DKRLVOC has the highest accuracy. Also, DKRLVOC has the highest g-mean for all 11 datasets and the highest precision for 7 datasets. In case of recall values, DKRLVOC achieves the highest recall for 8 out of 11 datasets, respectively. The efficiency of DKRLVOC is evident from the observation that it performs overwhelmingly better than other methods by scoring the highest g-mean, accuracy, precision, and recall for 11,8,7,8 datasets, respectively.

For OCC methods, the decision criteria is set during training time by taking a portion of data as outliers to improve its effectiveness to classify outliers. We present the variation of F_1 scores of the KELM methods and DKRLVOC for different small-size biomedical datasets across different percentage of dismissal, namely, δ = 1%, 5%, 10%, in Fig.2. In the figure, it can be observed that mostly for δ = 1%, DKRLVOC performs better than the other methods.

4.1.2. Experiments on other small-size datasets

We present the F_1 scores of 3 other small-size datasets in Table 4. DKRLVOC obtains the highest F_1 score on all 3 datasets as compared to the other OCC methods. DKRLVOC scores 1.28% $\sim$ 3.95% higher than the single-layer based methods and 5.87% $\sim$ 7.95% higher than the multi-layer based method, ML-OCKELM. Fig.3 shows the recall, precision, g-mean and accuracy comparisons between DKRLVOC and other KELM methods. It is quite clear from Fig.3, that for all 3 datasets DKRLVOC has the highest accuracy, g-mean and precision. For 2 datasets, DKRLVOC has the highest recall. We present the variation of F_1 scores of the KELM methods and DKRLVOC for other small-size datasets across different values of δ, namely, 1%, 5%, 10%, in Fig.4. In the figure, it can be observed that mostly for δ = 10%, DKRLVOC performs better than the other methods.

In the Table 4, DKRLVOC obtains an average F_1 score of 77.89 in comparison to OCKELM, VOCKELM, and ML-OCKELM, which obtain an average score of 72.92, 72.84 and 75.40, respectively. The better performance of DKRLVOC against OCKELM for biomedical data is attributed to the fact that DKRLVOC minimizes the intra-class variance in the first layer and uses multiple AE layers to extract relevant features from input. DKRLVOC enjoys the advantage of multiple reconstruction-based layers over VOCKELM, that reconstructs essential features at each layer, leading to better classification results. Also, the minimization of intra-class variance at first layer, puts DKRLVOC at an advantage over ML-OCKELM, leading to better results for biomedical datasets.

When comparing methods, runtime complexity is a crucial performance metric. The training time spent on different methods is recorded in Table 6. Due to the multi-layer architecture, the training time cost of DKRLVOC is

S.no.	Datasets	C	$C^{(j)}$	λ	k	δ
Biomedical datasets						
1	Arrhythmia	0.03125	1	1	3	0.01
2	Biomed	0.5	1	1	1	0.1
3	Breast Cancer	1	4	1	4	0.01
4	Caesarian	0.03125	0.03125	1	2	0.1
5	Cancer	0.25	4	1	10	0.01
6	Cardiotocography	32	16	1	1	0.05
7	Colposcopy	8	0.5	1	2	0.1
8	Cryotherapy	0.125	16	1	1	0.01
9	Hepatitis	0.25	0.0625	1	10	0.1
10	SPECT Heart	1	0.03125	1	1	0.01
11	Survival	0.125	0.03125	1	3	0.05
Other datasets						
12	Glass Building	0.0625	16	1	3	0.1
13	Ionosphere	0.25	0.125	1	1	0.1
14	Iris	8	8	1	1	0.01

Table 5: DKRLVOC parameters selected by 5-fold cross-validation for small-size UCI datasets.

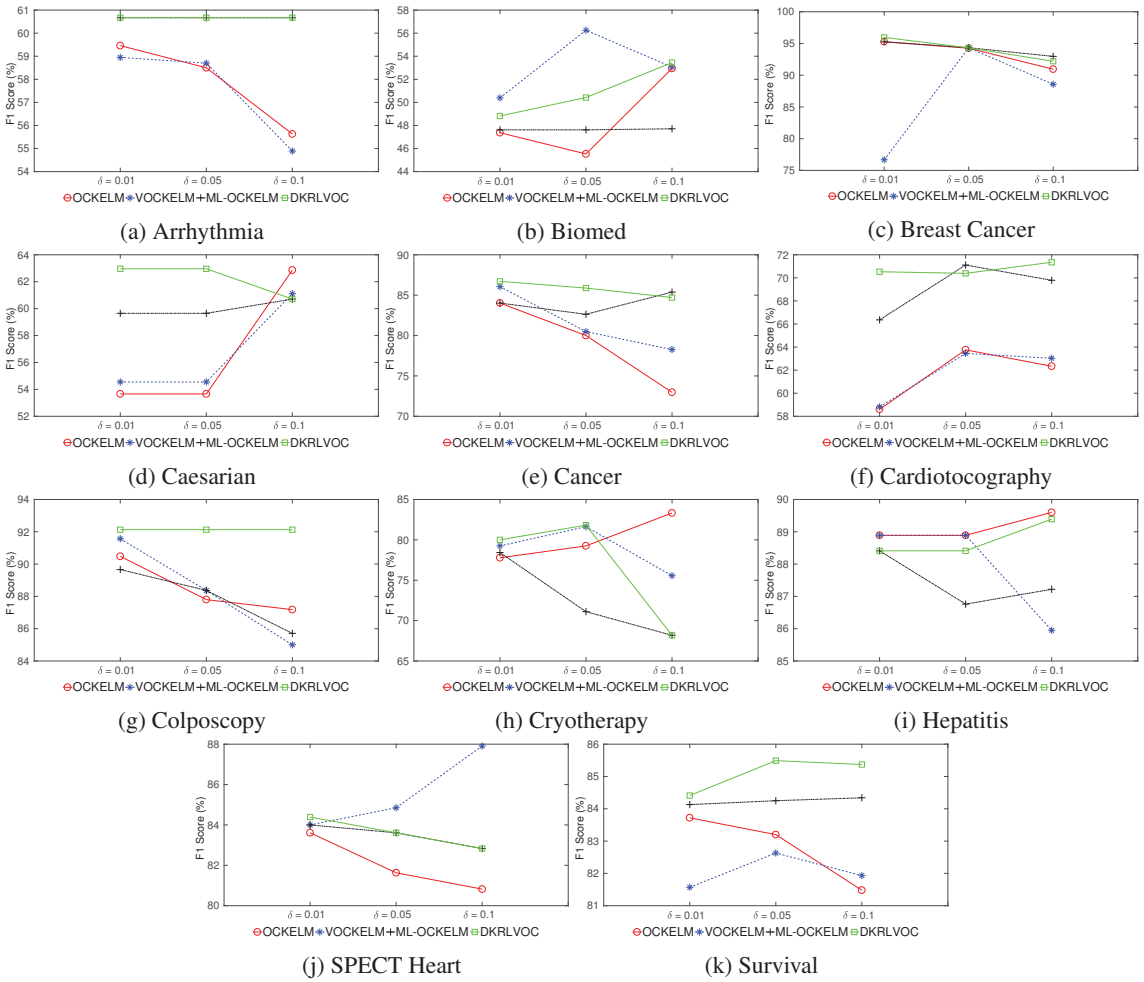


Figure 2: Examination of F_1 score for various percentage of dismissal for small-size biomedical datasets.

generally higher than the single-layer one-class methods, OCKELM and VOCKELM.

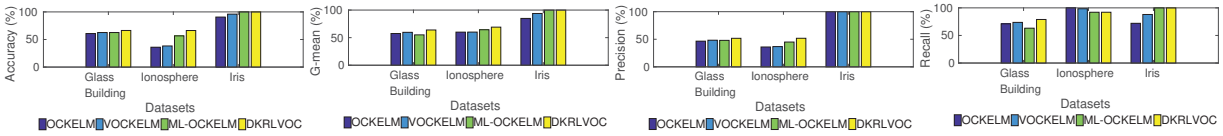
4.2. Experiments on medium-size datasets

We conduct experiments on 4 medium-size UCI datasets (2 biomedical and 2 other datasets). We provide the specifications of the two medium-size biomedical datasets in Table 7. The other two medium-size datasets, Optical digits, and Concordia handwritten digits, further consist of 10 classes each. We conduct experiments iteratively by taking each of the ten classes as target and the rest of the classes as outliers. The specifications of Optical digits and Concordia handwritten digits datasets are detailed in Table 8 and 9, respectively. The division of data between 5-fold cross-validation and testing is kept the same as small-size datasets.

Table 10, 12, 14 provides the optimal set of DKRLVOC parameters for the biomedical, optical digits, and concordia digits datasets, respectively. The parameters are selected using 5-fold cross-validation during training time.

S.no.	Datasets	OCKELM [31]	VOCKELM [32]	ML-OCKELM [33]	DKRLVOC
	Biomedical datasets				
1	Arrhythmia	0.038835	0.035595	0.027343	0.024965
2	Biomed	0.003331	0.010296	0.004299	0.009607
3	Breast Cancer	0.022137	0.045793	0.030913	0.052584
4	Caesarian	0.001849	0.003327	0.001741	0.003746
5	Cancer	0.003044	0.007889	0.012418	0.014098
6	Cardiotocography	0.00845	0.011117	0.009247	0.01292
7	Colposcopy	0.002229	0.004038	0.003184	0.004952
8	Cryotherapy	0.001216	0.003077	0.0015	0.003486
9	Hepatitis	0.00209	0.004601	0.00389	0.006503
10	SPECT Heart	0.005302	0.015587	0.013686	0.020492
11	Survival	0.00498	0.01043	0.009131	0.017571
	Other datasets				
12	Glass Building	0.00189	0.003598	0.002261	0.004432
13	Ionosphere	0.002542	0.009285	0.004699	0.007388
14	Iris	0.001386	0.004926	0.001745	0.005378

Table 6: Training time (in secs) on the small-size UCI datasets.



(a) Accuracy (b) G-mean (c) Precision (d) Recall

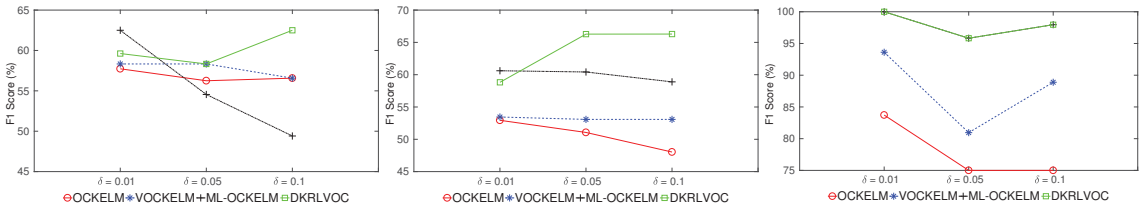
Figure 3: Examination of Accuracy, G-mean, Precision and Recall for different other small-size datasets.

We provide the F_1 scores for the two medium-size biomedical datasets, for DKRLVOC against different KELM methods in Table 11. The finest results are noted in bold. DKRLVOC achieves best F_1 score for Epileptic Seizure dataset while OCKELM performs best for Thyroid dataset. We compare and present the performance of different KELM methods based on recall, precision, g-mean and accuracy for both the datasets in the Fig.6. We present the variation of F_1 scores of the KELM methods and DKRLVOC for both medium-size biomedical datasets across different values of δ, namely, 1%, 5%, 10%, in Fig.7. It can be observed in the figure, that the curve for the classifiers mostly decreases with an increase in δ, suggesting improved performance of all the classifiers when the value of δ is low. Also, mostly for δ = 5% and 10%, DKRLVOC performs better than the other methods.

We provide the F_1 score for different target classes of optical digits dataset in Table 13. In Table, the best results are highlighted in bold. DKRLVOC obtains the highest F_1 score for 7 out of 10 classes, while VOCKELM and ML-OCKELM get the highest F_1 score for the remaining 1,2 classes, respectively. When we compare the average F_1 score over all the classes for each of the KELM methods, we can infer that the average F_1 score for DKRLVOC is the highest in comparison to the other methods. DKRLVOC scores 0.37% $\sim$ 20.86% higher than the single-layer KELM methods

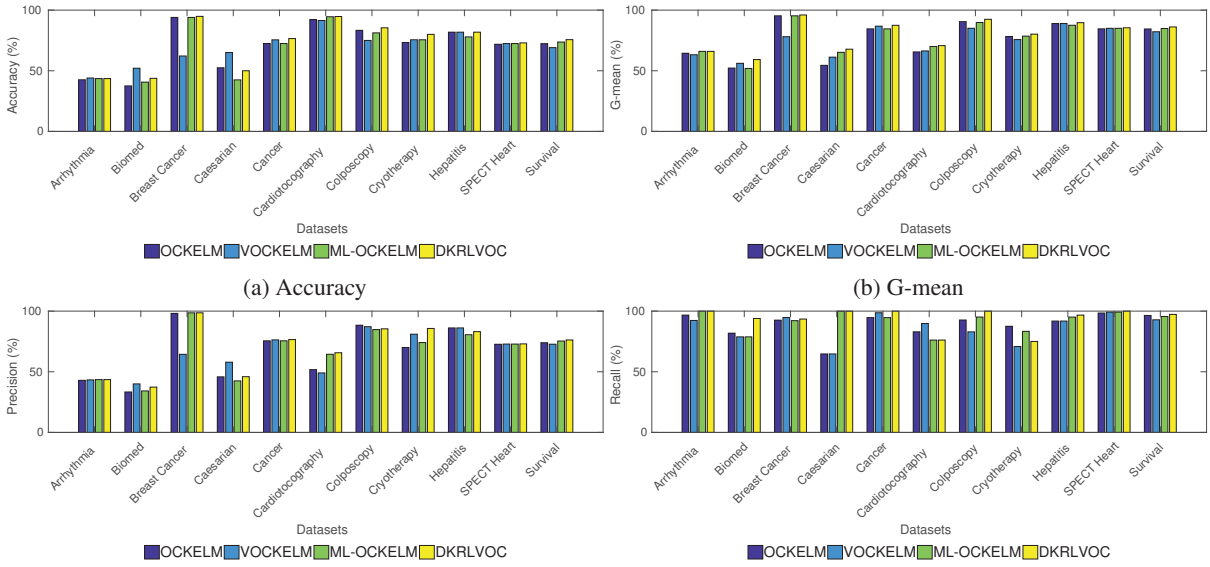
Datasets	#Total Samples	#Target	#Outlier	#Features	Target Class
Epileptic Seizure	4600	2300	2300	178	Normal (eyes open)
Thyroid	3772	3488	284	21	Subnormal

Table 7: UCI medium-size biomedical dataset specifications.



(a) Glass Building (b) Ionosphere (c) Iris

Figure 4: Examination of F_1 score for various percentage of dismissal for other small-size datasets.

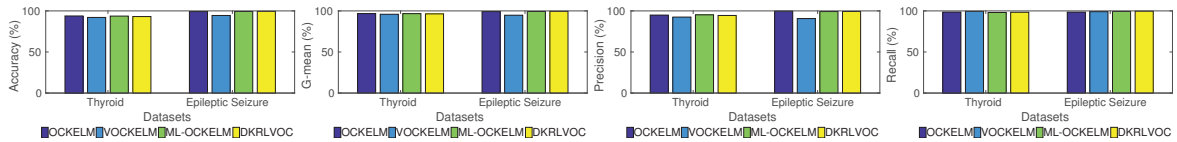


(a) Accuracy (b) G-mean

(c) Precision (d) Recall

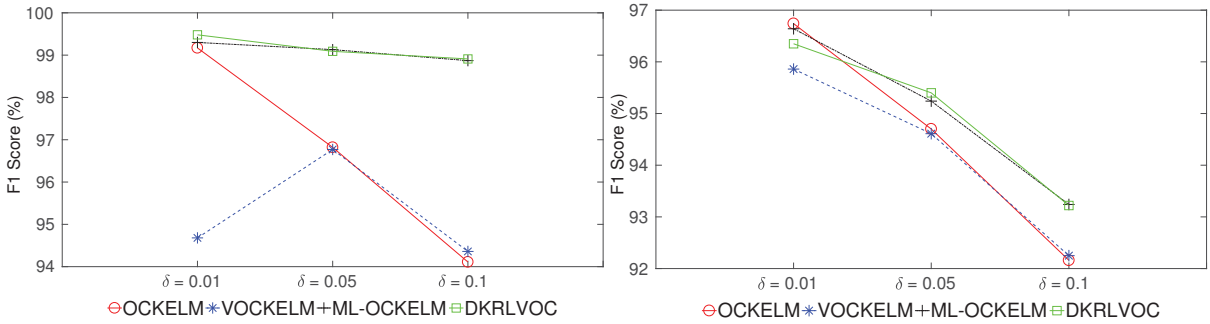
Figure 5: Examination of Accuracy, G-mean, Precision and Recall for different small-size biomedical datasets.

(OCKELM and VOCKELM), and 0.02% ~ 8.17% higher than the multi-layer KELM method (ML-OCKELM), across the 10 classes of optical digits dataset. From the above observations, it can be agreed that DKRLVOC performs better than the other KELM methods in terms of F_1 score. Comparisons have also been made based on accuracy, g-mean, precision, and recall in the Fig.8. DKRLVOC has the highest accuracy and g-mean for 7,7 classes, respectively. While DKRLVOC achieves the highest precision for 5 datasets, it certainly performs better in case of recall by achieving the highest recall values for 7 out of 10 datasets. It can be concluded that the proposed method performs far better than the other methods in case of accuracy, g-mean, and recall. We present the variation of F_1 scores of the KELM methods and DKRLVOC for all the classes of optical digits across different values of δ, namely, 1%, 5%, 10%, in Fig.9. It can be observed in the figure that across different values of δ, mostly for δ = 10%, DKRLVOC performs better than the other methods. Also, for 7 out of 10 cases, DKRLVOC curve decreases with an increase in the value of δ, suggesting improved performance of DKRLVOC when the value of δ is low. Also, the multi-layer methods perform better than



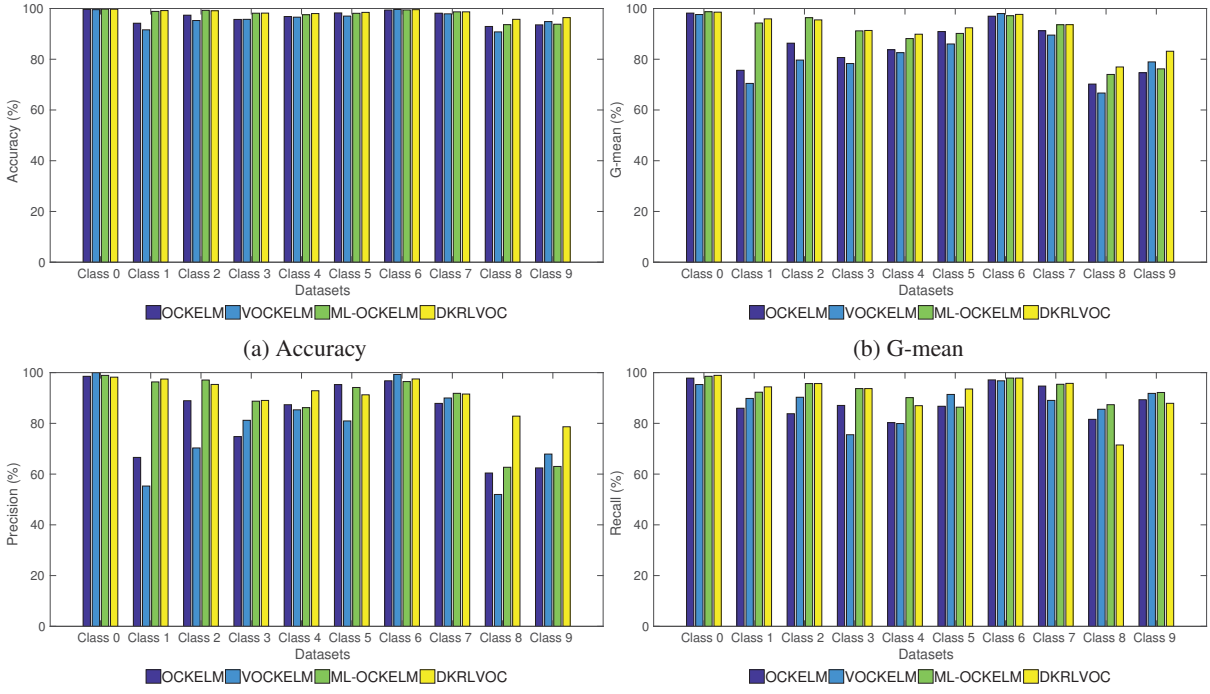
(a) Accuracy (b) G-mean (c) Precision (d) Recall

Figure 6: Examination of Accuracy, G-mean, Precision and Recall for different medium-size biomedical datasets.



(a) Epileptic Seizure (b) Thyroid

Figure 7: Examination of F_1 score for various percentage of dismissal for medium-size biomedical datasets.



(a) Accuracy (b) G-mean

(c) Precision (d) Recall

Figure 8: Examination of Accuracy, G-mean, Precision and Recall for different classes of optical digits dataset.

the single-layer methods in most cases.

We conduct experiments on concordia handwritten digits dataset by iteratively taking one of the 10 classes as target and the remaining 9 classes as outliers. In the concordia dataset, the number of outlier (negative class) samples is far greater than the number of target (positive class) samples as evident in Table 9. It can be observed in Fig.10 that DKRLVOC performs quite well on specificity but relatively poor on recall for the concordia dataset. Regardless, accuracy is found to be quite high, especially for classes 3 and 4. As discussed in section 4, it becomes quite evident from Fig.10 that in case of imbalanced data, accuracy fails to give a proper estimate of the performance of a model.

We present the F_1 scores obtained from the experiments for different target classes of concordia handwritten digits dataset in Table 15 with the best results highlighted in bold. DKRLVOC is found to get the highest F_1 score for 6 out of 10 classes of concordia dataset. The average F_1 score calculated over all the classes for each of the methods is found to be the highest for DKRLVOC as can be seen in Table 15. DKRLVOC scores $1.19\% \sim 26.07\%$ higher than the single-layer KELM methods, and $0.18\% \sim 5.8\%$ higher than the multi-layer KELM method, ML-OCKELM. It can be concluded that DKRLVOC performs far better than the other methods. We present the recall, precision, g-mean and accuracy for each case in fig 11. The method DKRLVOC scores the highest accuracy, g-mean and precision for 6,5,5 classes, respectively. We present the variation of F_1 score for all the classes of concordia digits across different values of δ, namely, 1%, 5%, 10%, in Fig.12. It can be observed that mostly for $\delta = 1\%$, DKRLVOC performs better than the other methods. For 8 out of 10 cases, the curves of single-layer and multi-layer methods are widely separated, clearly suggesting the efficiency of multi-layer classifiers over single-layer classifiers.

4.3. Experiments on real-world biomedical images

We present the analysis for experiments on MRI image data of Alzheimer's disease in section 4.3.1 and histopathological image data of breast cancer in section 4.3.2.

Target Class	#Target	#Outlier	#Features
0	554	5066	64
1	571	5049	64
2	557	5063	64
3	572	5048	64
4	568	5052	64
5	558	5062	64
6	558	5062	64
7	566	5054	64
8	554	5066	64
9	562	5058	64

Table 8: Specifications of UCI Dataset Optical digits.

Target Class	#Target	#Outlier	#Features
0	400	3600	256
1	400	3600	256
2	400	3600	256
3	400	3600	256
4	400	3600	256
5	400	3600	256
6	400	3600	256
7	400	3600	256
8	400	3600	256
9	400	3600	256

Table 9: Specifications of Concordia digits dataset.

S.no.	Datasets	C	$C^{(f)}$	λ	k	δ
1	Epileptic Seizure	2	0.5	1	3	0.01
2	Thyroid	0.0625	0.125	1	2	0.01

Table 10: DKRLVOC parameters selected by 5-fold cross-validation for medium-size biomedical datasets.

4.3.1. Alzheimer's disease classification

All MRI data used in this work were downloaded from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). ADNI was launched in the year 2003 as a public-private partnership, led by Principal Investigator Michael W. Weiner, MD. The main objective of ADNI is to analyze the use of neuroimaging techniques like MRI, positron emission tomography (PET), other biological markers, and clinical neuropsychological tests to estimate the onset of Alzheimer's disease from the state of mild cognitive impairment. For more information, visit `www.adni-info.org`.

We downloaded 100 T1-weighted structural MRI images (sMRI) from the ADNI database comprising of 50 control normal (CN) and 50 Alzheimer's disease (AD) subjects. The sMRI images of CN and AD subjects are shown in Fig.13. One can see the degeneration of neurons in Fig.13b for AD subject. In our dataset, the variation of subjects' age is in the range of 60-90 with a mean age of 75.83, and standard deviation of 6.07. For analysis of volume and thickness measures of the brain, the images were processed using Freesurfer software (version 6.0.1) recon-all pipeline [53, 54]. We included 34 cortical thickness measures, 23 subcortical tissue volumes, and 34 WM tissue volumes of each image. The volumetric data is normalized for variation in head size with division by total intracranial volume (TIV). For reporting F_1 scores in Table 18, the train-test split ratio is fixed at 80%-20%, meaning, 80% of target and outlier class data is used for 5-fold cross-validation and 20% is used as test set. The value of the regularization parameter $C^{(q)}$ for layer $q = 1, 2$ and $C^{(f)}$ for final layer is selected from the set $\{2^{-3}, 2^{-2}, ..., 2^{3}\}$ using 5-fold cross-validation. We provide the dataset specifications for Alzheimer's disease in Table 16. Table 17 provides the optimal set of DKRLVOC parameters for Alzheimer's disease datasets. The parameters are selected using 5-fold cross-validation during training time.

We compare our proposed DKRLVOC method with OCSVM, SVDD, OCKELM, VOCKELM, and ML-

	Thyriod	Epileptic Seizure
OCKELM [31]	**96.74**	99.17
VOCKELM [32]	95.86	94.68
ML-OCKELM [33]	96.64	99.3
DKRLVOC	96.35	**99.48**

Table 11: Comparisons of F_1 scores for medium-size biomedical datasets.

S.no.	Datasets	C	$C^{(f)}$	λ	k	δ
1	Class 0	16	4	1	3	0.01
2	Class 1	0.5	0.5	1	5	0.1
3	Class 2	16	1	1	1	0.05
4	Class 3	8	2	1	10	0.01
5	Class 4	0.03125	4	1	7	0.01
6	Class 5	0.125	0.5	1	8	0.1
7	Class 6	16	2	1	4	0.05
8	Class 7	4	2	1	1	0.05
9	Class 8	2	1	1	2	0.1
10	Class 9	0.03125	4	1	2	0.01

Table 12: DKRLVOC parameters selected by 5-fold cross-validation for optical digits dataset.

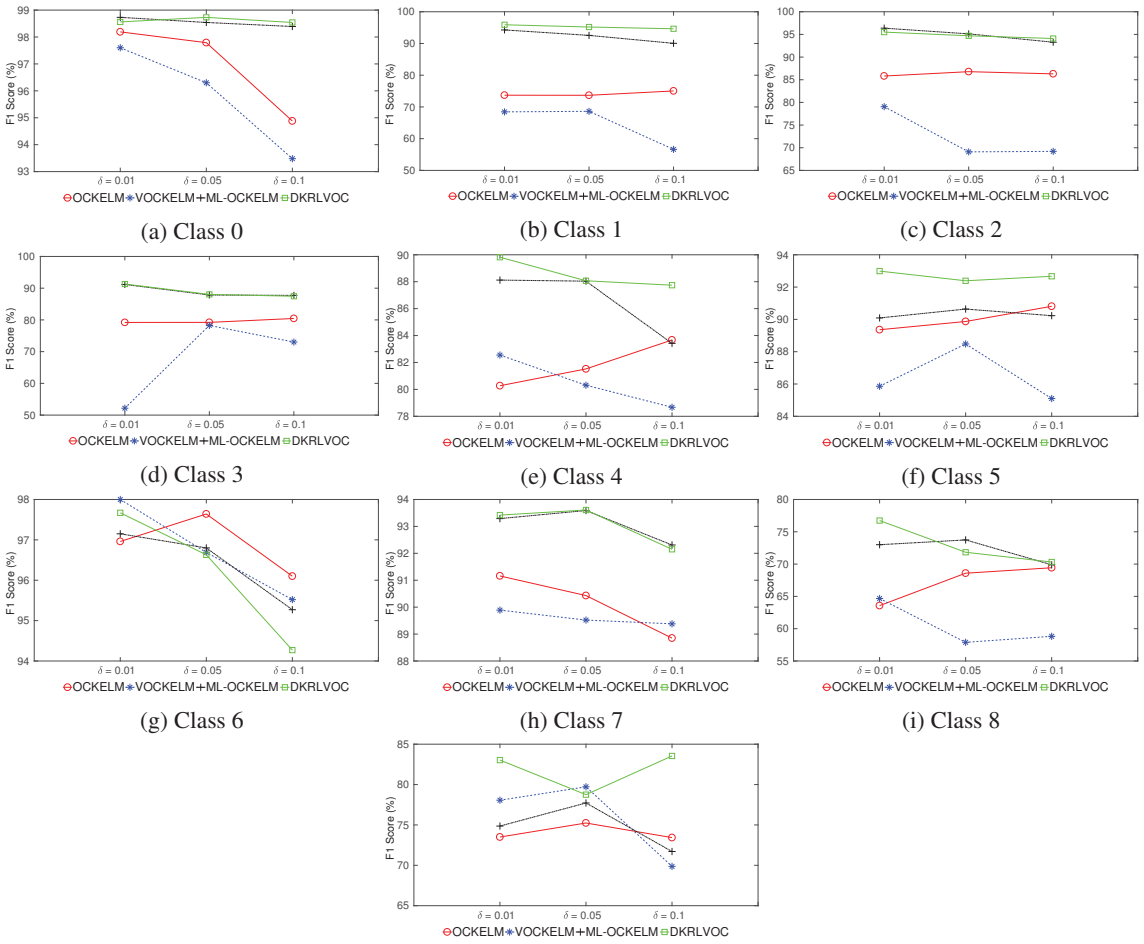
OCKELM on Alzheimer's data. We train the methods using CN data and treat AD as outliers. However, for comparison, we also present the results for training on AD class in Table 18 using F_1 score as the comparison metric. One can observe that for CN vs. AD case, DKRLVOC has F_1 score of 86.96% for cortical thickness and 81.82% for all features. It suggests that the cortical thickness and all features are prominent measures of MRI images for OCC. In Table 18 it can be observed that DKRLVOC performs better than the other methods for all four CN vs. AD cases. For AD vs. CN case, one can notice in Table 18 that the score is less than that of CN vs. AD in most cases. This may be attributed to the variation in neurodegeneration of AD patients. Therefore, the training of one-class based methods on CN data is more efficient than training on AD. In Table 18, DKRLVOC obtains an average F_1 score of 74.66, while OCKELM, VOCKELM, and ML-OCKELM score 70.84, 70.54 and 71.60, respectively. The multiple layers of AEs that reconstruct essential features at each layer put DKRLVOC at an advantage over OCKELM and VOCKELM. Also, the minimization of intra-class variance at first layer gives DKRLVOC an edge in performance over ML-OCKELM for Alzheimer's disease biomedical data.

In Fig.14, we present the variation in performance of different one-class methods when the amount of data available during training is varied. Following observations can be made from Fig.14:

1. DKRLVOC achieves better F_1 score in most train-test splits in the Fig.14a, 14c and 14d signifying that it does a better job in general than the other classifiers in identifying Alzheimer's in case of all features, subcortical

	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9	Average
OCKELM [31]	98.19	75.04	86.3	80.45	83.67	90.81	96.96	91.16	69.43	73.5	84.55
VOCKELM [32]	97.6	68.45	79.06	78.26	82.55	85.86	**98**	89.52	64.67	78.06	82.2
ML-OCKELM [33]	**98.73**	94.27	**96.38**	91.16	88.12	90.09	97.15	93.59	73	74.86	89.74
DKRLVOC	98.56	**95.9**	95.51	**91.31**	**89.82**	**92.39**	97.67	**93.61**	**76.74**	**83.03**	**91.45**

Table 13: Comparisons of F_1 scores for optical digits dataset.



(a) Class 0 (b) Class 1 (c) Class 2

(d) Class 3 (e) Class 4 (f) Class 5

(g) Class 6 (h) Class 7 (i) Class 8

(j) Class 9

Figure 9: Examination of F_1 score for various percentage of dismissal for different classes of optical digits dataset.

volume and white matter volume.

2. Fig.14a and 14b report better F_1 scores for all the methods over different train-test splits than the other two cases signifying all features and cortical thickness are prominent measures of Alzheimer's MRI images for OCC.

3. For 3 out of 4 cases, i.e., Fig.14b, 14c and 14d, a maximum F_1 score is reported at 80-20 train-test split for DKRLVOC. This is due to the fact that in 80-20 train-test split, more data is available during training.

4. The F_1 scores for all the methods are in close proximity to each other for lower train-test splits. This signifies that the performance of all the methods is quite similar when less training data is available.

Also, in Fig.14, it is noticeable that DKRLVOC is having high F_1 score values with less variation as compared to other methods. This shows that the performance of DKRLVOC is better and more stable for application of Alzheimer's

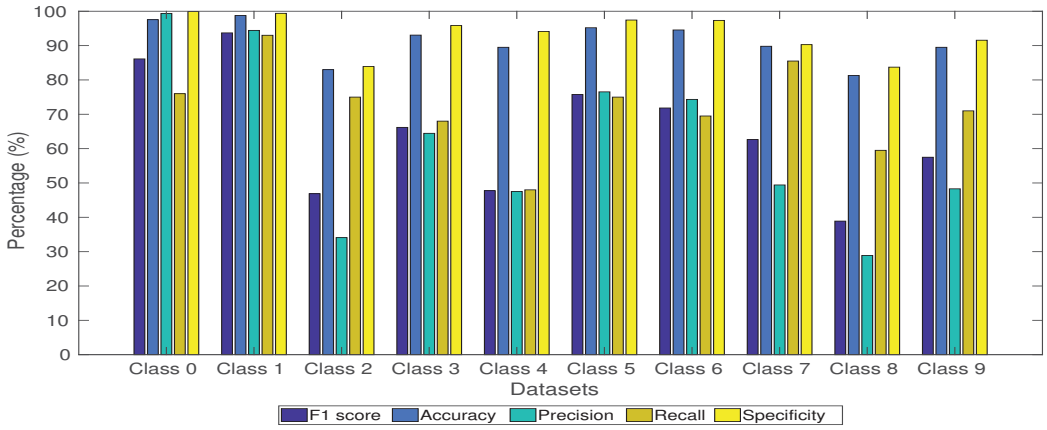
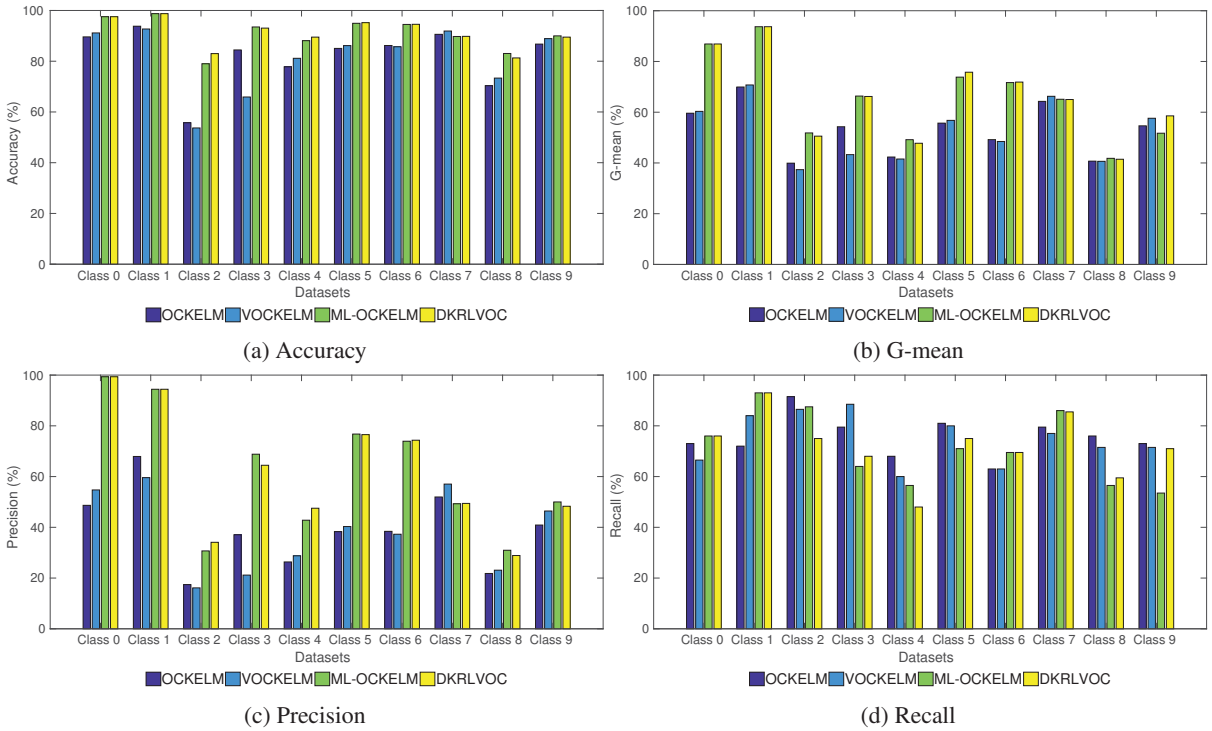


Figure 10: Comparison of different performance metrics on different classes of concordia dataset obtained using DKRLVOC method.



(a) Accuracy

(b) G-mean

(c) Precision

(d) Recall

Figure 11: Examination of Accuracy, G-mean, Precision and Recall for different classes of concordia dataset.

S.no.	Datasets	C	$C^{(f)}$	λ	k	δ
1	Class 0	8	2	1	2	0.01
2	Class 1	0.03125	4	1	3	0.01
3	Class 2	0.25	2	1	10	0.01
4	Class 3	4	8	1	7	0.01
5	Class 4	8	8	1	8	0.01
6	Class 5	2	8	1	6	0.01
7	Class 6	2	4	1	4	0.01
8	Class 7	16	2	1	10	0.05
9	Class 8	4	16	1	4	0.01
10	Class 9	0.25	2	1	2	0.01

Table 14: DKRLVOC parameters selected by 5-fold cross-validation for concordia digits dataset.

data. Apart from F_1 score, we utilize accuracy, g-mean, precision, and recall, as well, to show the efficiency of DKRLVOC over other one-class methods in Fig.15. The effectiveness of DKRLVOC is evident from the observation that out of 4 cases, DKRLVOC achieves the highest accuracy, g-mean, precision, and recall for 4, 4, 4, and 3 cases, respectively, for Alzheimer's disease.

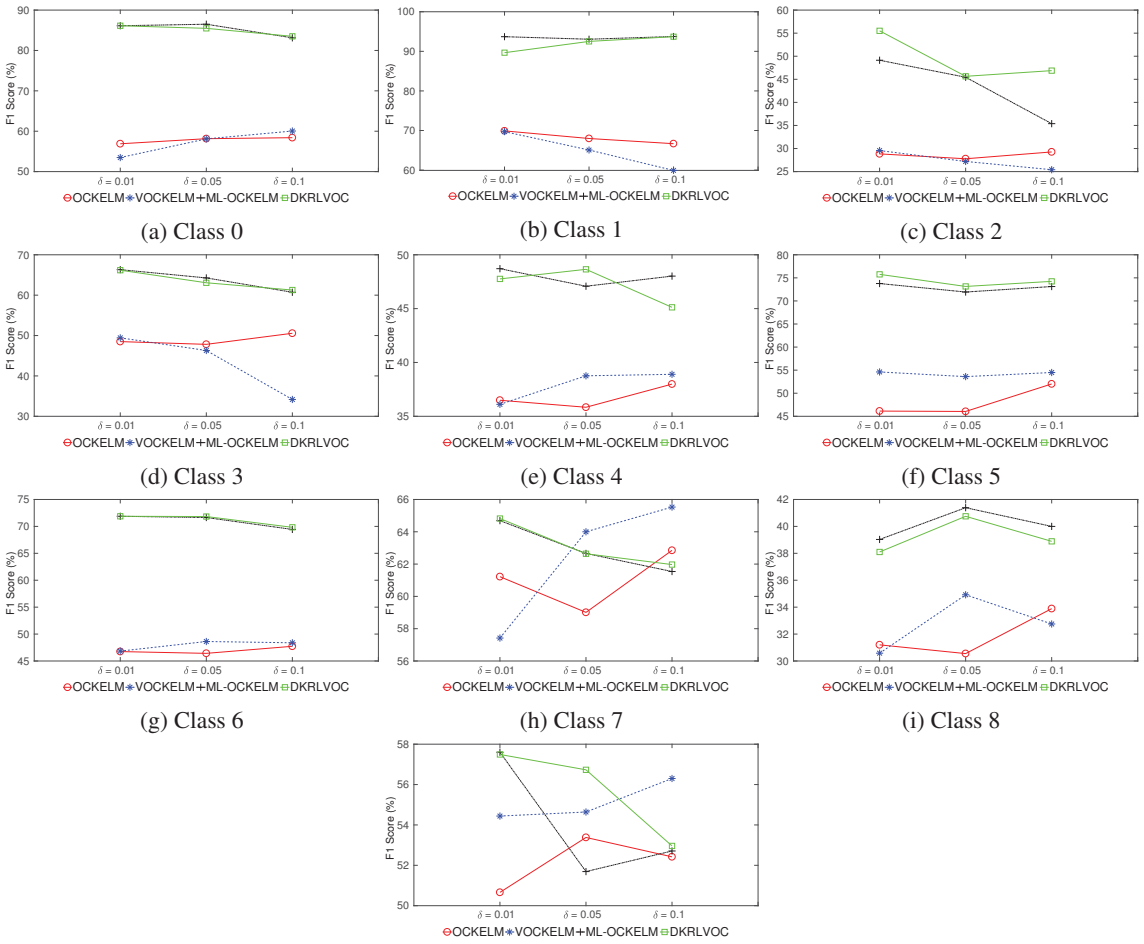
4.3.2. Breast cancer classification

For breast cancer, we use the BreakHis [55] histopathological image dataset. We use 1240 images from the dataset with 400X magnification. The images belong to two major categories: benign and malignant. The subclasses for the benign class are adenosis (AN), fibroadenoma (FA), phyllodes tumor (PT), and tubular adenoma (TA) having 106, 237, 115, and 130 images, respectively. In malignant class, the subclasses are ductal carcinoma (DC), lobular carcinoma (LC), mucinous carcinoma (MC), papillary carcinoma (PC) having 208, 137, 169, and 138 images, respectively. To extract useful features from the histopathological images, we convert the images into gray level and apply wavelet transform using Daubechies-4 (db4) wavelet up to 3 levels of decomposition [56, 57] as shown in Fig.16. The approximation and detail coefficients are concatenated to form the feature vector. The feature vectors are not normalized. For reporting F_1 scores in Table 21, the train-test split ratio is fixed at 80%-20%, meaning, 80% of target and outlier class data is used for 5-fold cross-validation and 20% is used as test set.

For making the distinction of the 4 subclasses of malignant cancer, we make 16 pairs of benign and malignant data, as shown in Table 19, while keeping the benign class as the target. The aim is to show the ability of DKRLVOC in differentiating the non-cancerous tumor from the cancerous tumor in possible pairs of benign and malignant class. Table 20 provides the optimal set of DKRLVOC parameters for Breast Cancer disease datasets. The parameters are selected using 5-fold cross-validation during training time.

	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9	Average
OCKELM [31]	58.4	69.9	29.26	50.56	37.99	52.01	47.73	62.85	33.89	52.42	49.501
VOCKELM [32]	60.05	69.71	27.2	34.17	38.9	53.6	46.84	**65.53**	34.92	56.3	48.722
ML-OCKELM [33]	**86.12**	**93.7**	45.45	**66.32**	**48.71**	73.77	71.65	62.66	**40**	51.69	64.007
DKRLVOC	**86.12**	**93.7**	**46.88**	66.18	47.76	**75.76**	**71.83**	62.64	38.89	**57.49**	**64.725**

Table 15: Comparisons of F_1 scores for concordia dataset.



(a) Class 0 (b) Class 1 (c) Class 2

(d) Class 3 (e) Class 4 (f) Class 5

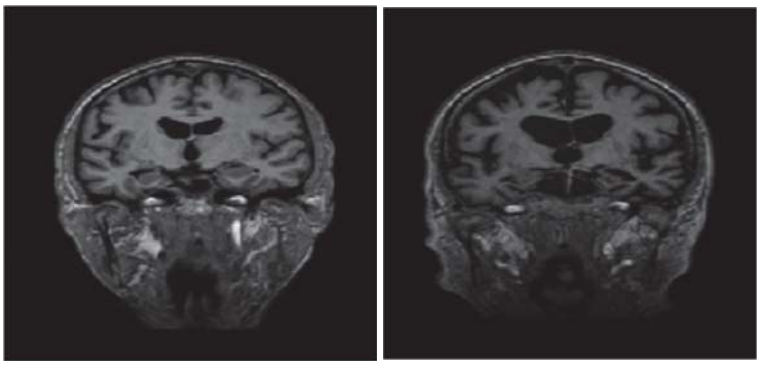
(g) Class 6 (h) Class 7 (i) Class 8

(j) Class 9

Figure 12: Examination of F_1 score for various percentage of dismissal for different classes of concordia dataset.

We show the comparison of DKRLVOC with the existing OCC methods in Table 21, using F_1 score as the performance metric. It is observable from Table 21, that DKRLVOC performs better than the other methods for 10 out of 16 breast cancer datasets. The highest F_1 score obtained is 81.03% for classification of fibroadenoma against papillary carcinoma. Moreover, the average F_1 score of DKRLVOC is the highest among all the methods. DKRLVOC achieves an average score of 69.41, while OCKELM, VOCKELM, and ML-OCKELM score 66.6, 67.96 and 66.97, respectively. The better performance of DKRLVOC over OCKELM and VOCKELM owe to the presence of multiple reconstruction-based layers. The reduction in intra-class variance at first layer helps in better separation of target class from outliers leading to improved performance of DKRLVOC over ML-OCKELM. These observations show the applicability of DKRLVOC on real-world biomedical datasets.

Additionally, we present the comparison of the different OCC methods on breast cancer in terms of accuracy,



(a) Control normal (CN) (b) Alzheimer's disease (AD)

Figure 13: MRI images of control normal and Alzheimer's disease subjects from ADNI database.

	Target Class	#Target	#Outlier	#Features
CN vs. AD	CN	50	50	91
AD vs. CN	AD	50	50	91

Table 16: Alzheimer's disease dataset specifications.

g-mean, precision, and recall in Fig.17. As evident from the Fig.17, DKRLVOC achieves highest accuracy, g-mean, precision, and recall against other OCC methods for 10, 11, 10 and 9 datasets, respectively. Moreover, we perform experiments over different train-test splits for breast cancer datasets in Fig.18. It shows that, for cases, where the number of samples in the target class is less, the F_1 score value over different train-test split is relatively less. When larger number of samples are available during training, as in Fig.18e, 18f, 18g, and 18h for target class FA, there is an improvement in the score along the x-axis.

5. Conclusion

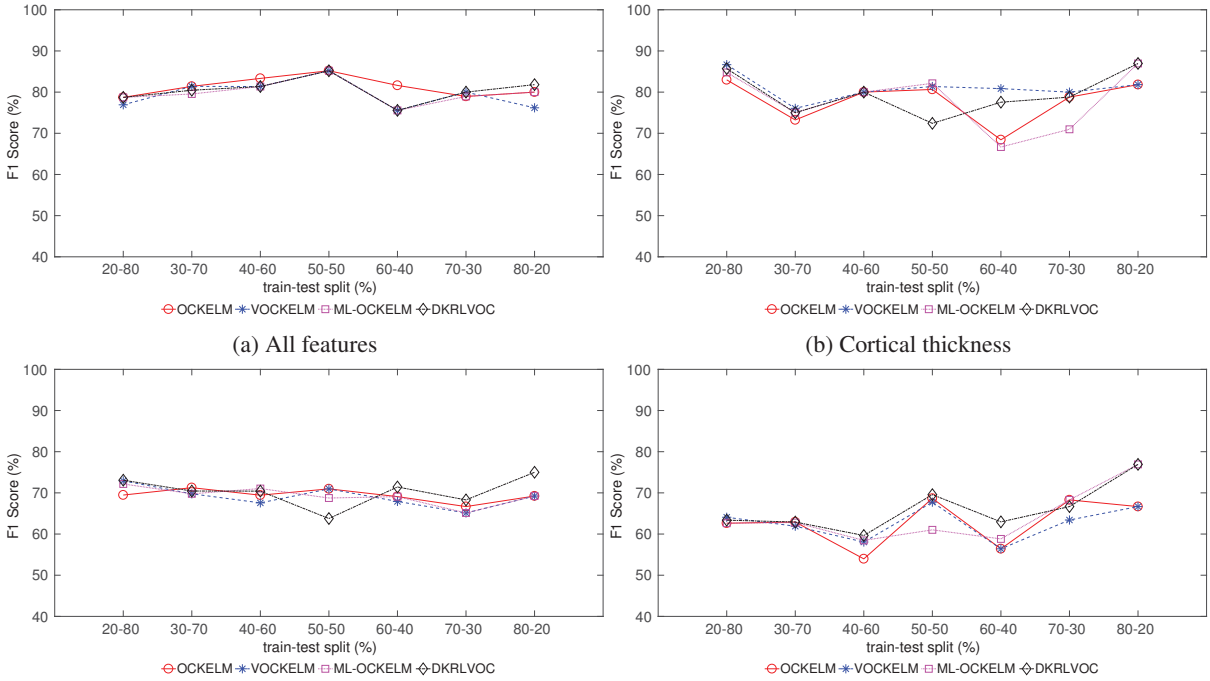
In this paper, we proposed a minimum variance-embedded deep KRL-based one-class classifier (DKRLVOC) for anomaly/ outlier detection. The proposed architecture comprises of multiple KRL based AEs and a final OCC layer. The stacked AEs enable better feature learning. The minimum variance embedding is done at the first layer, minimizing the intra-class variance which improves the generalization performance of the model. Increasing the number of training samples further improves the efficiency of first layer leading to better classification. To demonstrate

	CN vs. AD				AD vs. CN			
	All features	Cortical thickness	Subcortical volume	White matter volume	All features	Cortical thickness	Subcortical volume	White matter volume
C	0.5	0.125	0.5	0.5	0.25	0.25	0.25	0.125
Cf	2	0.125	2	0.125	8	2	0.125	0.125
lambda	1	1	1	1	1	1	1	1
k	3	7	9	3	2	1	3	1
delta	1	10	10	1	10	5	5	1

Table 17: DKRLVOC parameters selected by 5-fold cross-validation for Alzheimer's disease datasets.

	CN vs. AD				AD vs. CN				
	All features	Cortical thickness	Subcortical volume	White matter volume	All features	Cortical thickness	Subcortical volume	White matter volume	Average
OCSVM [12]	70.59	81.82	74.07	66.67	53.85	64.29	**80**	40	66.41
SVDD [13]	62.5	77.78	69.57	66.67	58.33	61.54	75	43.48	64.36
OCKELM [31]	80	81.82	69.23	66.67	**66.67**	**68.97**	66.67	**66.67**	70.84
VOCKELM [32]	76.19	81.82	69.23	66.67	62.07	66.67	75	**66.67**	70.54
ML-OCKELM [33]	80	**86.96**	69.23	**76.92**	62.07	66.67	64.29	**66.67**	71.60
DKRLVOC	**81.82**	**86.96**	**75**	**76.92**	64	**68.97**	76.92	**66.67**	**74.66**

Table 18: Comparisons of F_1 scores for Alzheimer's disease dataset.



(a) All features

(b) Cortical thickness

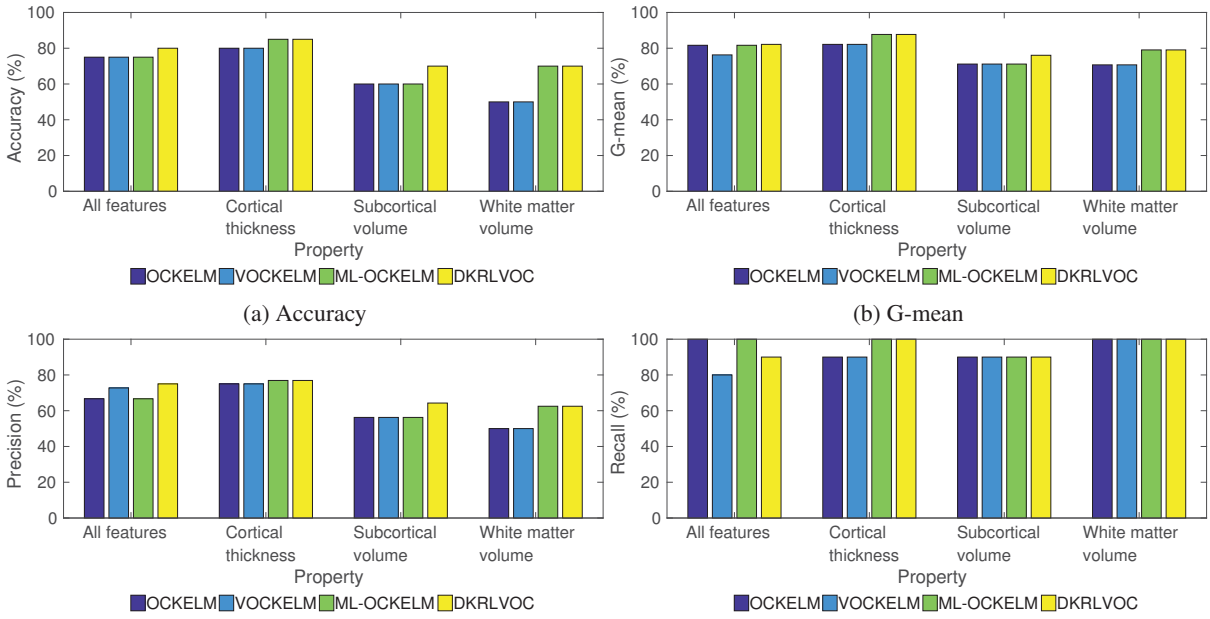
(c) Subcortical volume

(d) White matter volume

Figure 14: Examination of variation of F_1 score over different train-test split for Alzheimer's disease dataset.

the capability of DKRLVOC, we conducted experiments on 18 UCI benchmark datasets (13 biomedical and 5 other). The experimental results on biomedical datasets are summarized as follows,

- **Small-size datasets:** DKRLVOC obtained the highest F_1 score on all 11 small-size biomedical datasets. It performed 0.1% $\sim$ 6.77% better than the single-layer based classifiers and 0.39% $\sim$ 5.74% better than multi-layer based classifier.
- **Medium-size datasets:** The F_1 score obtained by DKRLVOC is 99.48 for Epileptic Seizure dataset, while OCKELM, VOCKELM, and ML-OCKELM scored 99.17, 94.68 and 99.3, respectively.
- **Real-world datasets:** For Alzheimer's disease dataset, DKRLVOC performs better than the other methods for all 4 CN vs. AD cases of Alzheimer's disease and obtained the highest F_1 score of 86.96% in comparison to



(a) Accuracy (b) G-mean

(c) Precision (d) Recall

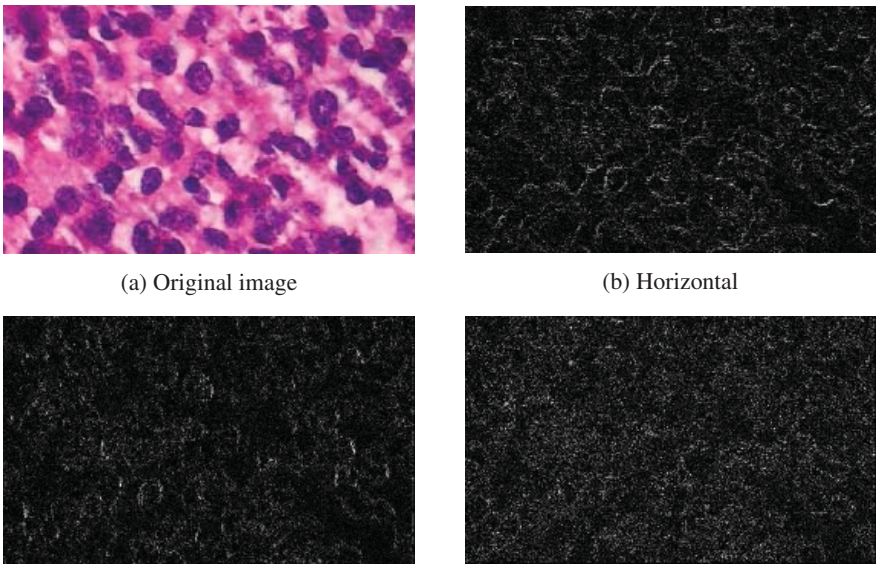
Figure 15: Examination of Accuracy, G-mean, Precision and Recall for different cases of Alzheimer's disease dataset.

	Target Class	Outlier Class	#Target	#Outlier	#Features
AN Vs DC	Adenosis	Ductalcarcinoma	106	208	768
AN Vs LC	Adenosis	Lobularcarcinoma	106	137	768
AN Vs MC	Adenosis	Mucinouscarcinoma	106	169	768
AN Vs PC	Adenosis	Papillarycarcinoma	106	138	768
FA Vs DC	Fibroadenoma	Ductalcarcinoma	237	208	768
FA Vs LC	Fibroadenoma	Lobularcarcinoma	237	137	768
FA Vs MC	Fibroadenoma	Mucinouscarcinoma	237	169	768
FA Vs PC	Fibroadenoma	Papillarycarcinoma	237	138	768
PT Vs DC	Phyllodes tumor	Ductalcarcinoma	115	208	768
PT Vs LC	Phyllodes tumor	Lobularcarcinoma	115	137	768
PT Vs MC	Phyllodes tumor	Mucinouscarcinoma	115	169	768
PT Vs PC	Phyllodes tumor	Papillarycarcinoma	115	138	768
TA Vs DC	Tubular adenoma	Ductalcarcinoma	130	208	768
TA Vs LC	Tubular adenoma	Lobularcarcinoma	130	137	768
TA Vs MC	Tubular adenoma	Mucinouscarcinoma	130	169	768
TA Vs PC	Tubular adenoma	Papillarycarcinoma	130	138	768

Table 19: Breast Cancer disease datasets specifications. Here, AN, FA, PT, TA, DC, LC, MC, PC refer to Adenosis, Fibroadenoma, Phyllodes tumor, Tubular adenoma, Ductalcarcinoma, Lobularcarcinoma, Mucinouscarcinoma, and Papillarycarcinoma, respectively.

existing methods. On application to breast cancer images, DKRLVOC performed better than the other methods for 10 out of 16 breast cancer datasets and achieved the highest F_1 score of 81.03% for classification of fibroadenoma and papillary carcinoma.

Coming to other datasets, DKRLVOC scored 1.28% $\sim$ 3.95% higher than the single-layer based methods and 5.87% $\sim$ 7.95% higher than the multi-layer based method. For optical digits dataset, DKRLVOC scored 0.37% $\sim$ 20.86% higher than the single-layer methods, and 0.02% $\sim$ 8.17% higher than the multi-layer method. For concordia digits dataset, DKRLVOC scored 1.19% $\sim$ 26.07% higher than the single-layer methods, and 0.18% $\sim$ 5.8% higher



(a) Original image (b) Horizontal

(c) Vertical (d) Diagonal

Figure 16: Histopathological image of (a) ductal carcinoma. Image of different detail coefficients obtained after wavelet transform on image (a) are shown in subfigures (b)-(d).

than the multi-layer method.

Above discussion justifies that the proposed deep kernel method performs quite well in case of small-size datasets. Since collecting relevant data is quite expensive or time taking process in most of the biomedical field, DKRLVOC is quite apt for this field. Moreover, the computational time is also reduced owing to the non-iterative nature of DKRLVOC.

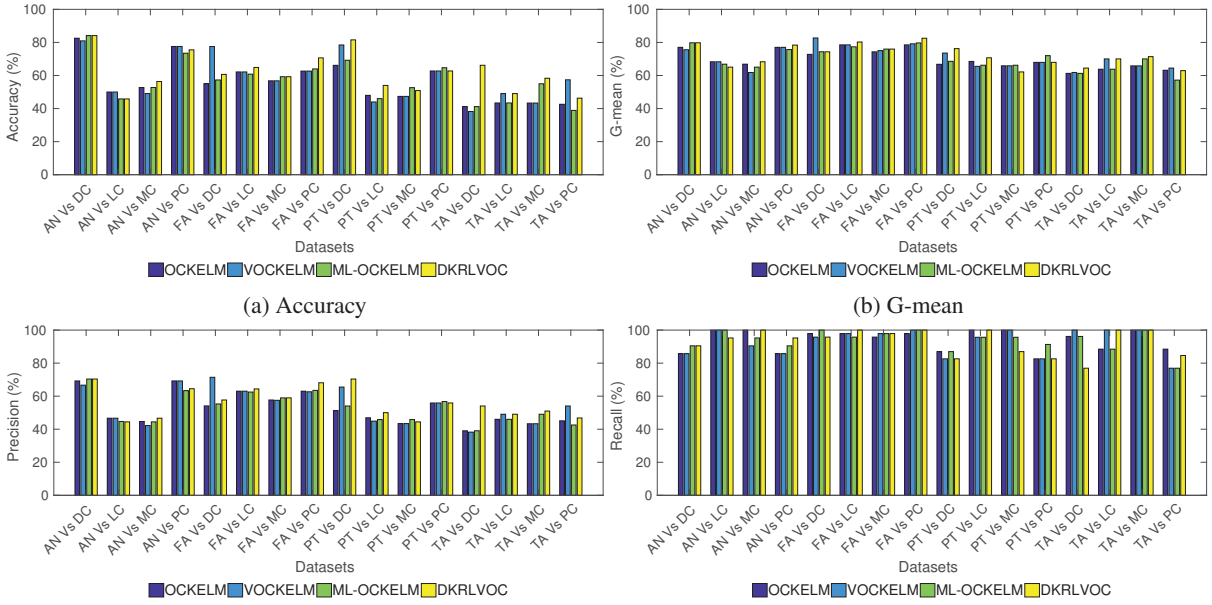
The performance of DKRLVOC improves with an increase in data available during training. However, with an increase in training data, the computation of the inverse of a matrix during the calculation of output weight becomes increasingly difficult. In the future, further research can be conducted in order to tackle the difficulty of calculation of

S.no.	Datasets	C	$C^{(f)}$	λ	k	δ
1	AD Vs DC	0.5	4	1	3	0.01
2	AD Vs LC	2	1	1	4	0.01
3	AD Vs MC	0.125	2	1	10	0.05
4	AD Vs PC	1	0.0625	1	10	0.05
5	FA Vs DC	1	4	1	1	0.1
6	FA Vs LC	0.03125	2	1	3	0.01
7	FA Vs MC	0.03125	0.25	1	6	0.01
8	FA Vs PC	4	0.25	1	9	0.01
9	PD Vs DC	0.25	32	1	4	0.05
10	PD Vs LC	0.25	0.03125	1	1	0.05
11	PD Vs MC	0.0625	0.5	1	4	0.05
12	PD Vs PC	0.25	0.03125	1	8	0.1
13	TA Vs DC	32	1	1	3	0.05
14	TA Vs LC	0.03125	8	1	2	0.01
15	TA Vs MC	0.25	16	1	3	0.05
16	TA Vs PC	2	4	1	2	0.05

Table 20: DKRLVOC parameters selected by 5-fold cross-validation for Breast Cancer disease datasets.

	OCSVM [12]	SVDD [13]	OCKELM [31]	VOCKELM [32]	ML-OCKELM [33]	DKRLVOC
AN Vs DC	79.07	71.43	76.6	75	**79.17**	**79.17**
AN Vs LC	58.06	49.12	**63.64**	**63.64**	61.76	60.61
AN Vs MC	**64.41**	50.98	61.76	57.58	60.61	63.64
AN Vs PC	72.73	66.67	76.6	76.6	74.51	**76.92**
FA Vs DC	67.2	66.12	69.7	**81.82**	71.21	72
FA Vs LC	74.58	72.07	76.67	76.67	75.63	**78.33**
FA Vs MC	71.19	67.86	72	72.44	**73.6**	**73.6**
FA Vs PC	75.68	72.9	76.67	77.05	77.69	**81.03**
PT Vs DC	59.7	65.57	64.52	73.08	66.67	**76**
PT Vs LC	57.58	56.25	63.89	61.11	61.97	**66.67**
PT Vs MC	52.94	49.23	60.53	60.53	**61.97**	58.82
PT Vs PC	66.67	61.82	66.67	66.67	**70**	66.67
TA Vs DC	52.87	54.76	55.56	55.32	55.56	**63.49**
TA Vs LC	53.52	52.17	60.53	**65.82**	60.53	**65.82**
TA Vs MC	51.85	51.85	60.47	60.47	65.82	**67.53**
TA Vs PC	59.46	62.86	59.74	**63.49**	54.79	60.27
Average	63.59	60.73	66.6	67.96	66.97	**69.41**

Table 21: Comparisons of F_1 scores for Breast Cancer disease dataset.



(a) Accuracy (b) G-mean

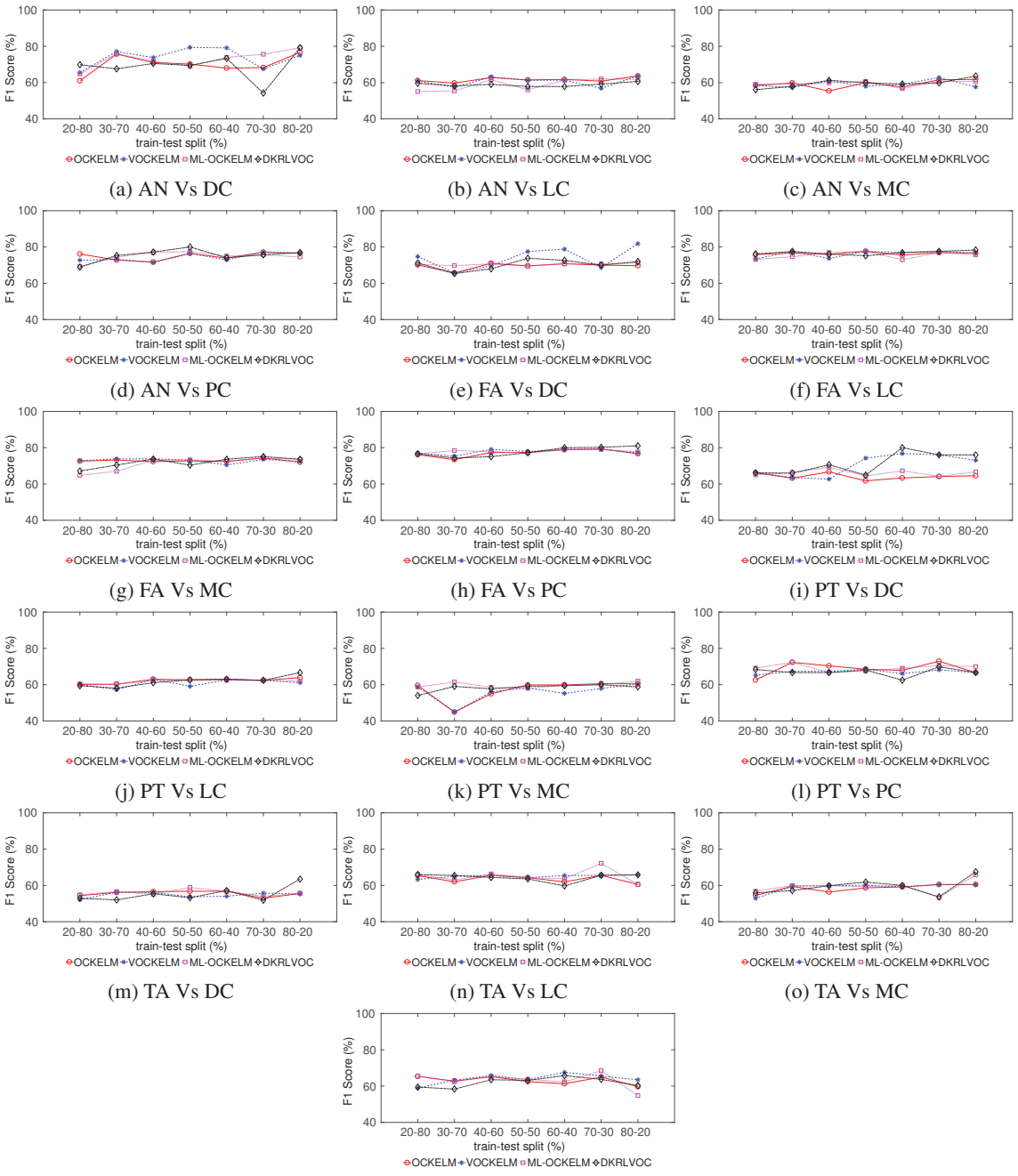
(c) Precision (d) Recall

Figure 17: Examination of Accuracy, G-mean, Precision and Recall for Breast Cancer disease datasets.

inverse for a large number of samples. Additionally, while the proposed method is capable of handling only stationary data, further research can be done to extend DKRLVOC to handle online streaming data and non-stationary data.

Acknowledgements

We are thankful to the Editor and anonymous reviewers for their valuable feedback and constructive comments for the improvement of this paper. This work is supported by Department of Science and Technology, INDIA under Ramanujan fellowship scheme grant no. SB/S2/RJN-001/2016 and Council of Scientific & Industrial Research (CSIR),



(a) AN Vs DC (b) AN Vs LC (c) AN Vs MC

(d) AN Vs PC (e) FA Vs DC (f) FA Vs LC

(g) FA Vs MC (h) FA Vs PC (i) PT Vs DC

(j) PT Vs LC (k) PT Vs MC (l) PT Vs PC

(m) TA Vs DC (n) TA Vs LC (o) TA Vs MC

(p) TA Vs PC

Figure 18: Examination of variation of F_1 score over different train-test split for Breast Cancer disease datasets.

New Delhi, INDIA under Extra Mural Research (EMR) scheme grant no. 22(0751)/17/EMR-II.

The collection of data and sharing of this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904), and DOD ADNI (Department of Defense award

number W81XWH-12-2-0012). The funding for ADNI is provided by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: AbbVie, Alzheimers Association; Alzheimers Drug Discovery Foundation; Araclon Biotech; BioClinica, Inc.; Biogen; Bristol-Myers Squibb Company; CereSpir, Inc.; Cogstate; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Lumosity; Lundbeck; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Takeda Pharmaceutical Company; and Transition Therapeutics. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (`www.fnih.org`). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimers 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.

References

[1] J. Mourão-Miranda, D.R. Hardoon, T. Hahn, A.F. Marquand, S.C. Williams, J. Shawe-Taylor, and M. Brammer. Patient classification as an outlier detection problem: an application of the one-class support vector machine. *Neuroimage*, 58(3):793–804, 2011.

[2] H.J. Shin, D.H. Eom, and S.S. Kim. One-class support vector machines-an application in machine fault detection and classification. *Computers & Industrial Engineering*, 48(2):395–408, 2005.

[3] L. Manevitz and M. Yousef. One-class document classification via neural networks. *Neurocomputing*, 70(7-9):1466–1481, 2007.

[4] G. Cohen, M. Hilario, H. Sax, S. Hugonnet, C. Pellegrini, and A. Geissbuhler. An application of one-class support vector machine to nosocomial infection detection. *Studies in health technology and informatics*, 107(Pt 1):716–720, 2004.

[5] C.P. Diehl and J.B. Hampshire. Real-time object classification and novelty detection for collaborative video surveillance. In *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*, volume 3, pages 2620–2625. IEEE, 2002.

[6] M. Markou and S. Singh. A neural network-based novelty detector for image sequence analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1664–1677, 2006.

[7] D.M.J. Tax. One-class classification: Concept learning in the absence of counter-examples. 2002.

[8] L. Tarassenko, P. Hayton, N. Cerneaz, and M. Brady. Novelty detection for the identification of masses in mammograms. In *1995 Fourth International Conference on Artificial Neural Networks*, pages 442–447. IET, 1995.

[9] E. Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[10] A. Ypma and R.P. Duin. Support objects for domain approximation. In *International Conference on Artificial Neural Networks*, pages 719–724. Springer, 1998.

[11] E.M. Knorr, R.T. Ng, and V. Tucakov. Distance-based outliers: algorithms and applications. *The VLDB JournalThe International Journal on Very Large Data Bases*, 8(3-4):237–253, 2000.

[12] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[13] D.M. Tax and R.P. Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004.

[14] M.F. Jiang, S.S. Tseng, and C.M. Su. Two-phase clustering process for outliers detection. *Pattern recognition letters*, 22(6-7):691–700, 2001.

[15] G.A. Carpenter, S. Grossberg, and D.B. Rosen. Art 2-a: An adaptive resonance algorithm for rapid category learning and recognition. *Neural networks*, 4(4):493–504, 1991.

[16] C.M. Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.

[17] N. Japkowicz, C. Myers, M. Gluck, et al. A novelty detection approach to classification. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 518–523, 1995.

[18] J. Hertz, R.G. Palmer, and A.S. Krogh. *Introduction to the Theory of Neural Computation*. Perseus Publishing, 1st edition, 1991.

[19] P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.

[20] E. Pekalska, D.M. Tax, and R. Duin. One-class lp classifiers for dissimilarity representations. In *Advances in neural information processing systems*, pages 777–784, 2003.

[21] Chesner Désir, Simon Bernard, Caroline Petitjean, and Laurent Heutte. One class random forests. *Pattern Recognition*, 46(12):3490–3506, 2013.

[22] Markos Markou and Sameer Singh. Novelty detection: a reviewpart 1: statistical approaches. *Signal Processing*, 83(12):2481 – 2497, 2003.

[23] Markos Markou and Sameer Singh. Novelty detection: a reviewpart 2:: neural network based approaches. *Signal Processing*, 83(12):2499 – 2521, 2003.

[24] Marco A.F. Pimentel, David A. Clifton, Lei Clifton, and Lionel Tarassenko. A review of novelty detection. *Signal Processing*, 99:215 – 249, 2014.

[25] S.S. Khan and M.G. Madden. One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review*, 29(3):345–374, 2014.

[26] M.A. Pimentel, D.A. Clifton, L. Clifton, and L. Tarassenko. A review of novelty detection. *Signal Processing*, 99:215–249, 2014.

[27] R.P.W. Duin. On the choice of smoothing parameters for parzen estimators of probability density functions. *IEEE Transactions on Computers*, (11):1175–1179, 1976.

[28] P. Juszczak, D.M. Tax, E. Pe, R.P. Duin, et al. Minimum spanning tree based one-class classifier. *Neurocomputing*, 72(7-9):1859–1869, 2009.

[29] D.S. Hochbaum and D.B. Shmoys. A best possible heuristic for the k-center problem. *Mathematics of operations research*, 10(2):180–184, 1985.

[30] L.E. Ghaoui, M.I. Jordan, and G.R. Lanckriet. Robust novelty detection with single-class mpm. In *Advances in neural information processing systems*, pages 929–936, 2003.

[31] Q. Leng, H. Qi, J. Miao, W. Zhu, and G. Su. One-class classification with extreme learning machine. *Mathematical problems in engineering*, 2015, 2015.

[32] V. Mygdalis, A. Iosifidis, A. Tefas, and I. Pitas. One class classification applied in facial image analysis. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1644–1648. IEEE, 2016.

[33] H. Dai, J. Cao, T. Wang, M. Deng, and Z. Yang. Multilayer one-class extreme learning machine. *Neural Networks*, 2019.

[34] Yungang Zhang, Bailing Zhang, Frans Coenen, Jimin Xiao, and Wenjin Lu. One-class kernel subspace ensemble for medical image classification. *EURASIP Journal on Advances in Signal Processing*, 2014(1):17, 2014.

[35] S. Dreiseitl, M. Osl, C. Scheibböck, and M. Binder. Outlier detection with one-class svms: an application to melanoma prognosis. In *AMIA*

Annual Symposium Proceedings, volume 2010, page 172. American Medical Informatics Association, 2010.

[36] G. Iordanescu, P.N. Venkatasubramanian, and A.M. Wyrwicz. Automatic segmentation of amyloid plaques in mr images using unsupervised support vector machines. *Magnetic resonance in medicine*, 67(6):1794–1802, 2012.

[37] Rongling Lang, Ruibo Lu, Chenqian Zhao, Honglei Qin, and Guodong Liu. Graph-based semi-supervised one class support vector machine for detecting abnormal lung sounds. *Applied Mathematics and Computation*, 364:124487, 2020.

[38] Jianguo Zhang, Kai-Kuang Ma, Meng-Hwa Er, and Vincent Chong. Tumor Segmentation from Magnetic Resonance Imaging by Learning via one-class support vector machine. In *International Workshop on Advanced Image Technology (IWAIT '04)*, pages 207–211, Singapore, Singapore, January 2004.

[39] J. Zhou, K. L. Chan, V. F. H. Chong, and S. M. Krishnan. Extraction of brain tumor from mr images using one-class support vector machine. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 6411–6414, Jan 2005.

[40] R. KHUTLANG, S. KRISHNAN, A. WHITELAW, and T. S. DOUGLAS. Automated detection of tuberculosis in ziehl-neelsen-stained sputum smears using two one-class classifiers. *Journal of Microscopy*, 237(1):96–102, 2010.

[41] K. Cuppens, P. Karsmakers, A. Van de Vel, B. Bonroy, M. Milosevic, S. Luca, T. Croonenborghs, B. Ceulemans, L. Lagae, S. Van Huffel, and B. Vanrumste. Accelerometry-based home monitoring for detection of nocturnal hypermotor seizures based on novelty detection. *IEEE Journal of Biomedical and Health Informatics*, 18(3):1026–1033, May 2014.

[42] Xin Bi, He Ma, Jianhua Li, Yuliang Ma, and Deyang Chen. A positive and unlabeled learning framework based on extreme learning machine for drug-drug interactions discovery. *Journal of Ambient Intelligence and Humanized Computing*, Aug 2018.

[43] Chandan Gautam, Aruna Tiwari, Sundaram Suresh, and Alexandros Iosifidis. Multi-layer kernel ridge regression for one-class classification. *CoRR*, abs/1805.07808, 2018.

[44] Christina P. The state of the art of dementia research: New frontiers. *World Alzheimer's Report 2018*, 2018.

[45] X. Bi and H. Wang. Early alzheimers disease diagnosis based on eeg spectral images using deep learning. *Neural Networks*, 2019.

[46] G. Lee, K. Nho, B. Kang, K.A. Sohn, and D. Kim. Predicting alzheimers disease progression using multi-modal deep learning approach. *Scientific reports*, 9(1):1952, 2019.

[47] S. Liu, S. Liu, W. Cai, S. Pujol, R. Kikinis, and D. Feng. Early diagnosis of alzheimer's disease with deep learning. In *2014 IEEE 11th international symposium on biomedical imaging (ISBI)*, pages 1015–1018. IEEE, 2014.

[48] M. Tanveer, B. Richhariya, R.U. Khan, A.H. Rashid, P. Khanna, M. Prasad, and C.T. Lin. Machine learning techniques for the diagnosis of Alzheimer's disease: A review. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, (In press), 2019.

[49] Bernhard Schölkopf, Robert C Williamson, Alex J Smola, John Shawe-Taylor, and John C Platt. Support vector method for novelty detection. In *NIPS*, volume 12, pages 582–588, 1999.

[50] D.M. Tax and R.P. Duin. Data description in subspaces. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 2, pages 672–675. IEEE, 2000.

[51] D.M.J. Tax. Ddtools, the data description toolbox for matlab, Jan 2018. version 2.1.3.

[52] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[53] M. Reuter, N.J. Schmansky, H.D. Rosas, and B. Fischl. Within-subject template estimation for unbiased longitudinal image analysis. *NeuroImage*, 61(4):1402–1418, 2012.

[54] E. Westman, J.S. Muehlboeck, and A. Simmons. Combining MRI and CSF measures for classification of alzheimer's disease and prediction of mild cognitive impairment conversion. *NeuroImage*, 62(1):229–238, 2012.

[55] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte. A dataset for breast cancer histopathological image classification. *IEEE Transactions*

on Biomedical Engineering, 63(7):1455–1462, July 2016.

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

[57] H-G Hwang, H-J Choi, B-I Lee, H-K Yoon, S-H Nam, and H-K Choi. Multi-resolution wavelet-transformed image analysis of histological sections of breast carcinomas. *Analytical Cellular Pathology*, 27(4):237–244, 2005.