Generalized Reference Kernel With Negative Samples For Support Vector One-class Classification

Jenni Raitoharju
Faculty of Information Technology, *University of Jyväskylä*, Jyväskylä, Finland, 0000-0003-4631-9298

***Abstract*—This paper focuses on small-scale one-class classification with some negative samples available. We propose Generalized Reference Kernel with Negative Samples (GRKneg) for One-class Support Vector Machine (OC-SVM). We study different ways to select/generate the reference vectors and recommend an approach for the problem at hand. It is worth noting that the proposed method does not use any labels in the model optimization but uses the original OC-SVM implementation. Only the kernel used in the process is improved using the negative data. We compare our method with the standard OC-SVM and with the binary Support Vector Machine (SVM) using different amounts of negative samples. Our approach consistently outperforms the standard OC-SVM using Radial Basis Function kernel. When there are plenty of negative samples, the binary SVM outperforms the one-class approaches as expected, but we show that for the lowest numbers of negative samples the proposed approach clearly outperforms the binary SVM.**

***Index Terms*—One-class Support Vector Machine, negative samples, Generalized Reference Kernel**

I. Introduction

One-class classification techniques are used in situations where samples only from a single (positive) class are available, but it is necessary to create a model for recognizing outliers (negative samples) [1]. In recent years, deep learning based techniques, e.g., [2], have been proposed, but also the traditional techniques, such as One-class Support Vector Machine (OC-SVM) [3] and Support Vector Data Description (SVDD) [4], are still commonly used in small-scale problems [5], [6] and new variants and extensions [7], [8] are being proposed.

If information about outliers is available either via expert knowledge or via existence of some negative samples, it should be possible to produce better one-class classification models. This scenario has been extensively considered in large-scale deep learning-based outlier detection [9], [10], but it has received surprisingly little attention in connection to traditional one-class classification techniques. An explanation for this lack of attention could be that these techniques typically perform poorly with negative data. Therefore, when any negative data are available, either binary classifiers or their variants for imbalanced classification, e.g., [11], are used instead.

SVDD variant considering negative samples was proposed already along the original SVDD [4], but the authors concluded that using negative samples in the model optimization often leads to *worse* results than using only positive samples. More recent SVDD variants using negative samples [12], [13] have been able to boost the performance of the standard SVDD, but not to the level of binary Support Vector Machine (SVM). In [14], SVM and SVDD were combined to tackle the problem of one-class classification with negative samples. Extensions of OC-SVM to consider negative samples have been also proposed [15], [16], but the implementations are not publicly available. A larger number of one-class classification studies, such as [7], [17], has used negative samples for model validation (hyperparameter selection), while they are not used in the core model optimization. In such works, it is typically not evaluated how critical the availability of representative negative samples is for the model performance. Finally, some works have compared one-class and binary classification methods in different settings [18], [19].

In this paper, we propose a new method for small-scale one-class classification with negative samples. Our method is based on OC-SVM and Generalized Reference Kernel (GRK) [20]. We propose a novel approach for selecting GRK reference vectors based on the negative training samples in a manner that can significantly and consistently boost the method performance.

II. Generalized Reference Kernel with Negative Samples for OC-SVM

Generalized Reference Kernel (GRK) [20] has methodological similarities with approximate kernel approaches, such as random sampling methods [21], [22], random projection methods [23], and random Fourier features [24], [25]. However, the main idea of GRK is significantly different. The approximate kernel approaches are designed for large datasets, where the original kernel approaches are computationally too expensive. They aim at finding a computationally lighter way to approximate the original kernel matrix or function. GRK, on the other hand, was proposed for one-class classification tasks where the amount of data is low. Instead of trying to approximate the original kernel, GRK aims at providing a better kernel using implicit data augmentation via reference vectors used in the kernel computation. As the reference vectors do not need to be labeled, the problems of data augmentation in the absence of negative data can be avoided.

In [20], it was shown that selecting the reference vectors in GRK as the (positive) training samples augmented with random vectors generated from the training data distribution (assuming standard normal distribution for the standardized training set) could improve the results for SVDD. For OC-SVM, data augmentation by sampling reference vectors from the distribution of the positive data was not equally successful, but the initial results suggested that selecting negative samples as reference vectors could improved the results more. However,

TABLE I: GRK/GRKneg variants used in the experiments

Variant	Reference vectors
1	P positive training samples
2	P positive + N negative training samples
3	N negative training samples
4	P positive and N negative generated samples
5	$P+N$ negative generated samples
6	P non-positive and N negative generated samples
7	P negative generated samples and N negative training samples (Proposed)
8	$2P$ negative generated samples and N negative training samples
9	$P/2$ negative generated samples and N negative training samples

the potential of using GRK with negative samples was not further studied.

In this paper, we propose Generalized Reference Kernel with Negative Samples (GRKneg). The proposed kernel can be used instead of the regular, usually Radial Basis Function (RBF), kernel in the standard kernel OC-SVM, when some negative training samples are available. Our hypothesis is that if we have some negative samples, we can use them to generate more samples from the approximated negative class distribution and improve the OC-SVM performance by setting them as reference vectors for GRKneg. Indeed, our experiments presented in Section IV show that by setting the reference vectors as the negative samples augmented by vectors sampled from the approximated negative distribution, we can significantly boost the OC-SVM performance.

The full algorithm for computing the GRKneg matrix for OC-SVM with the proposed reference vector selection approach is provided in Algorithm 1. As in [20], we use a tilde to denote the base kernel and related terms used in the process. Derivations for the basic GRK operations, such as centering of the matrices, can be found in [20]. It is worth emphasizing that the final GRKneg kernel size is equivalent to the RBF kernel size, i.e., $P \times P$, where P is the number of positive training samples, irrespective of the number of reference vectors used in computing the kernel. The negative and generated samples are not used directly in the OC-SVM model optimization, but only to form a more discriminative representation for the positive training samples.

After forming the GRKneg kernel matrix, the original kernel OC-SVM implementation can be used for one-class classification. As it is a well-known method, we skip its details here.

III. EXPERIMENTAL SETUP

A. GRKneg/GRK Variants and Comparative Methods

To find the best way to use the negative training samples in GRKneg kernel computation, we compared different ways of selecting the reference vectors as summarized in Table I. Some of the compared approaches use also positive training samples as in the original GRK. Positive generated samples were randomly sampled from a normal distribution with zero mean and unit variance (the data have been standardized wrt. positive training data), negative generated samples were

Algorithm 1: Generalized Reference Kernel with Negative Samples (GRKneg) for OC-SVM

Training

Input : $\mathbf{P} = [\mathbf{p}_1, ..., \mathbf{p}_P] \in \mathbb{R}^{\mathcal{D}\times P}$, *%Pos. train data*
$\mathbf{N} = [\mathbf{n}_1, ..., \mathbf{n}_N] \in \mathbb{R}^{\mathcal{D}\times N}$, *%Neg. train data*
$\tilde{\kappa}(\cdot,\cdot)$, *%Base kernel function*

Output: $\mathbf{K_{PP}} \in \mathbb{R}^{P\times P}$, *%GRK matrix*

%Compute mean and std of negative samples
$\mu_{neg} = \mathrm{mean}(\mathbf{n}_1, \dots, \mathbf{n}_N)$,
$\sigma_{neg} = \mathrm{std}(\mathbf{n}_1, \dots, \mathbf{n}_N)$

%Sample P random vectors
$\mathbf{M} \in \mathbb{R}^{\mathcal{D}\times P} \sim \mathcal{N}(\mu_{neg}, \sigma^2_{neg})$

%Collect reference vectors to $\mathbf{R} \in \mathbb{R}^{\mathcal{D}\times(N+P)=\mathcal{D}\times R}$
$\mathbf{R} = [\mathbf{N}; \mathbf{M}]$

%Compute uncentered $\tilde{\mathbf{K}}_{\mathbf{RR}} \in \mathbb{R}^{R\times R}$ *and center it*
$[\tilde{\mathbf{K}}_{\mathbf{RR}}]_{ij} = \tilde{\kappa}(\mathbf{r}_i, \mathbf{r}_j)$
$\tilde{\mathcal{K}}_{\mathbf{RR}} = (\mathbf{I} - \frac{1}{R}\mathbf{1}_R\mathbf{1}_R^T)\tilde{\mathbf{K}}_{\mathbf{RR}}(\mathbf{I} - \frac{1}{R}\mathbf{1}_R\mathbf{1}_R^T)$

% Calculate the eigendecomposition $\tilde{\mathcal{K}}_{\mathbf{RR}}$
$\tilde{\mathcal{K}}_{\mathbf{RR}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$

%Compute the pseudoinverse of $\tilde{\mathcal{K}}_{\mathbf{RR}}$ using the r non-zero eigenvalues
$\tilde{\mathcal{K}}^+_{\mathbf{RR}} = \mathbf{U}_r\mathbf{\Lambda}_r^{-1}\mathbf{U}_r^T$,

%Compute uncentered $\tilde{\mathbf{K}}_{\mathbf{RP}} \in \mathbb{R}^{R\times P}$ *and center it*
$[\tilde{\mathbf{K}}_{\mathbf{RP}}]_{ij} = \tilde{\kappa}(\mathbf{r}_i, \mathbf{p}_j)$
$\tilde{\mathcal{K}}_{\mathbf{RP}} = (\mathbf{I} - \frac{1}{R}\mathbf{1}_R\mathbf{1}_R^T)\left(\tilde{\mathbf{K}}_{\mathbf{RR}} - \tilde{\mathbf{K}}_{\mathbf{RP}}(\frac{1}{R}\mathbf{1}_R\mathbf{1}_P^T)\right)$

%Compute the GRKneg matrix
$\mathbf{K_{PP}} = \tilde{\mathcal{K}}^T_{\mathbf{RP}}\tilde{\mathcal{K}}^+_{\mathbf{RR}}\tilde{\mathcal{K}}_{\mathbf{RP}} = \tilde{\mathcal{K}}_{\mathbf{PR}}\tilde{\mathcal{K}}^+_{\mathbf{RR}}\tilde{\mathcal{K}}_{\mathbf{RP}}$

Testing

Input : $\tilde{\mathcal{K}}_{\mathbf{PR}}, \tilde{\mathcal{K}}^+_{\mathbf{RR}}, \mathbf{R}$, *%Computed in training*
$\mathbf{Y} \in \mathbb{R}^{\mathcal{D}\times Y}$, *%Test data*

Output: $\mathbf{K_{PY}} \in \mathbb{R}^{P\times Y}$, *%GRK matrix for test data*

%Compute uncentered $\tilde{\mathbf{K}}_{\mathbf{RY}} \in \mathbb{R}^{R\times Y}$ *and center it*
$[\tilde{\mathbf{K}}_{\mathbf{RY}}]_{ij} = \tilde{\kappa}(\mathbf{r}_i, \mathbf{y}_j)$
$\tilde{\mathcal{K}}_{\mathbf{RY}} = (\mathbf{I} - \frac{1}{R}\mathbf{1}_R\mathbf{1}_R^T)\left(\tilde{\mathbf{K}}_{\mathbf{RR}} - \tilde{\mathbf{K}}_{\mathbf{RY}}(\frac{1}{R}\mathbf{1}_R\mathbf{1}_Y^T)\right)$

%Compute the GRKneg matrix for test data
$\mathbf{K_{PY}} = \tilde{\mathcal{K}}_{\mathbf{PR}}\tilde{\mathcal{K}}^+_{\mathbf{RR}}\tilde{\mathcal{K}}_{\mathbf{RY}}$

randomly sampled from a normal distribution having the same mean and variance as the negative samples as in Algorithm 1. Non-positive generated samples in variant 6 were generated first as positive samples and then 0.5 was added to the absolute value of each element to push the values further away from the origin. Here, the idea was to generate samples that are on the outskirts of the positive distribution without using the negative samples. We also compared our approach with the standard kernel OC-SVM algorithm and with binary SVM classifier.

B. Implementation and Computing Environment

We used in all our experiments Matlab R2017b and SVM implementation from LIBSVM library[1]. For binary SVM, we used the C-SVC (-s 0) and, for OC-SVM, the one-class SVM implementation (-s 2). To be able to use our custom kernel and to ensure similar hyperparameter setting in all the RBF kernels including those used as base kernels in GRK, we manually compute all the kernels (-t 4). Our implementation is available at https://github.com/JenniRaitoharju/GRKneg.

C. Datasets and Negative Sample Selection

For our experiments, we selected publicly available datasets from UCI Machine Learning Repository[2]. We created one-class classification tasks by considering each class as the positive class and the other class(es) as the negative class. A summary of the dataset properties is shown in Table II. We used 70% of each class for training and 30% for testing. This splitting was done 5 times. In all tasks, the data was standardized with respect to the positive training data. The hyperparameters were set using a cross-validation within the training set as further explained in Section III-D.

To study the impact of different amounts of negative data, we selected randomly, but keeping same selection for all methods, 5, 10, 20, 30 or all the negative samples to be used in the experiments. Thus, we had 14 different one-class classification tasks and, for each, 5 different subtasks with a different number of negative samples. Furthermore, each task and subtask was repeated 5 times over different train-test splittings.

D. Hyperparameter Selection and Evaluation Metric

We used 5-fold cross-validation within the training set to evaluate the hyperparameters separately for each subtask described in Section III-C. As we consider only scenarios, where at least few negative training samples are available for training, we avoided the common challenge of one-class classification, where the hyperparameter optimization should be done without negative data. As a part of our experimental setup, we also evaluate how much the number of negative samples affects the final performance of the standard OC-SVM, where the negative samples are used only for hyperparameter selection.

[1]https://www.csie.ntu.edu.tw/~cjlin/libsvm/

[2]http://archive.ics.uci.edu/ml

TABLE II: Datasets and subtasks used in the experiments

Dataset	C	N_{tot}	$\mathcal{D}$	Task abr.	Target class	P
Iris	3	150	4	Iris1	Setosa	35
				Iris2	Versicolor	35
				Iris3	Virginica	35
Seeds	3	210	7	Seed1	Kama	49
				Seed2	Rosa	49
				Seed3	Canadian	49
Ionosphere	2	351	32	Ion1	Good	157
				Ion2	Bad	88
Sonar	2	208	60	Son1	Rock	67
				Son2	Mines	77
Qualitative bankruptcy	2	250	6	Bank1	No bankr.	100
				Bank2	Bankr.	74
Somerville happiness	2	143	6	Happ1	Unhappy	46
				Happ2	Happy	53

C - number of classes, N_{tot}- total number of samples,
D - dimensionality, Task abr. - task abbreviation in other tables
P - number of positive training samples in task

In all methods and variants, we use the RBF kernel defined as

$$\tilde{\kappa}(\mathbf{p}_i, \mathbf{p}_j) = \exp\left(\frac{-\|\mathbf{p}_i - \mathbf{p}_j\|_2^2}{2\sigma^2}\right), \quad (1)$$

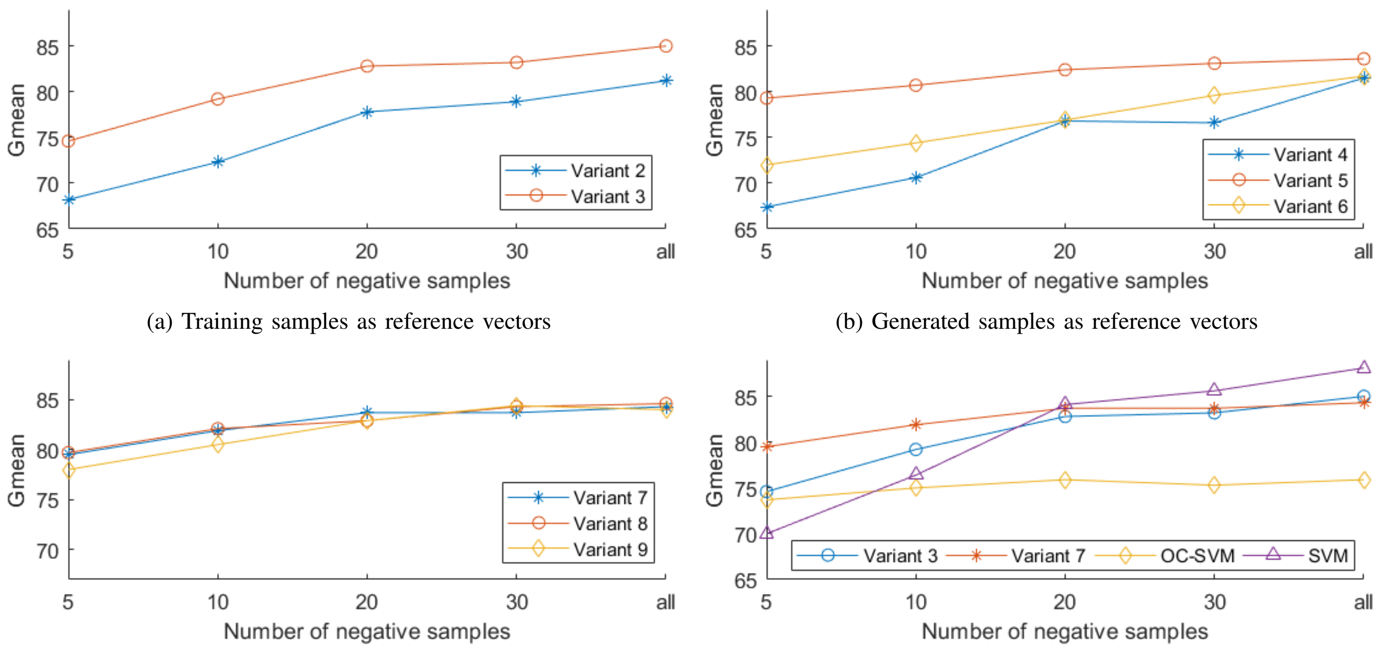
where σ is a hyperparameter. In OC-SVM and SVM, the RBF kernel was used directly as the main kernel, while GRKneg/GRK variants used it as the base kernel for computing the generalized kernel. We set σ in (1) to be $\sqrt{sd_{aver}}$, where d_{aver} was the average squared distance been the training samples and s was selected from $s = \{10^{-1}, 10^0, 10^1, 10^2, 10^3\}$. The C for C-SVC implementation was selected from $C = \{10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3\}$ and the ν for OC-SVM approaches from $\nu = \{0.05, 0.1, 0.15, 0.2\}$.

As our evaluation metric, we used Geometric Mean (Gmean), because it considers both True Positive Rate (TPR) and True Negative Rate (TNR). Gmean is defined as Gmean $= \sqrt{\text{TPR} \times \text{TNR}}$.

IV. Experimental Results

Fig. 1 shows the average Gmean scores over all the one-class classification tasks with different numbers of negative training samples. Figs. 1a-c show results of OC-SVM with different GRK/GRKneg variants described in Table I. Fig. 1-d compares the variants 3 and 7 with the standard OC-SVM and binary SVM. Below, we discuss the results shown in each subfigure.

Fig. 1-a: How do different training samples work as reference vectors? When only positive training samples were used as reference vectors (variant 1), the performance was so poor (Gmean below 50) that we do not even show the curve in order to keep the same scale for all the subfigures. This poor performance is in accordance with the results of [20]. The variant 2 using both positive and negatives samples works significantly better. However, the variant 3 using only negative samples, even when there are only 5 of them, consistently outperforms the other variants. We can conclude that that using positive training samples as reference vectors in GRK for OC-SVM is not recommendable.



(a) Training samples as reference vectors

(b) Generated samples as reference vectors

(c) Both training and generated samples as reference vectors

(d) Comparisons with standard methods

Fig. 1: Average Gmean values over all one-class classification tasks

Fig. 1-b: If we generate reference vectors instead of using the actual training vectors, how are the results affected? From which distribution should we generate the samples? As we know the importance of having reference vectors from the negative distribution, variants 4-6 all have N negative generated samples and, in addition, P positive, negative, or non-negative generated samples. We see that having negative generated samples (variant 5) outperforms the other options. Having non-negative generated samples (variant 6) is better than having positive generated samples (variant 4), but not as good as having negative generated samples. On the hand, approximating the negative distribution with a normal distribution having the same mean and variance must be a really coarse approximation in most cases. We experimented also with some other ways to generate negative samples. For examples, we tried adding noise to the negative training samples. However, we could not find anything better than the coarse normal distribution approximation.

Fig. 1-c: Is it better to have only generated negative samples or a combination of negative training and generated samples? How does the amount of generated samples affect the results? Variant 7 is close to variant 5, but N generated negative samples in variant 5 have been replaced with the negative training samples in variant 7. The difference between variants 5 and 7 is small and difficult to see by comparing subfigures b and c, but for all the amounts of negative samples the average Gmean is slightly higher for variant 7. Also the differences between having P (variant 7), $2P$ (variant 8), or $P/2$ (variant 9) negative generated samples is quite small. $P/2$ has a slightly worse performance for the lowest numbers of negative training samples, whereas P and $2P$ generated samples lead to very similar performances and, therefore, we pick the lower number. As a result of all the comparisons, our proposed approach is variant 7 that has P negative generated samples and N negative training samples.

Fig. 1-d: How are standard OC-SVM and binary SVM affected by the number of negative training samples? Can the proposed approach outperform these approaches? OC-SVM is affected by the number of negative training samples only via cross-validation used for hyperparameter selection. The impact is relatively small, but still the average Gmean varies from 73.7 for 5 negative samples to 75.9 for all negative samples. The impact on SVM is much larger and, with 5 negative samples its average Gmean is worse than for standard OC-SVM. The proposed variant 7 clearly outperforms OC-SVM on all numbers of negative samples and clearly outperforms SVM on 5 and 10 negative samples. On 30+ negative samples, the binary SVM outperforms all the one-class approaches, which is an expected result. Comparison of variants 3 and 7 shows that having both negative training and generated samples as reference vectors is most beneficial for the lowest numbers of negative training samples, whereas using only negative samples can be a better option when plenty of negative samples are available. As the target use case for the proposed method has only a low number of negative training data, this further confirms that variant 7 is our proposed GRKneg approach.

Table III shows task-specific numerical results for approaches compared in Fig. 1-d for 5, 10, and 20 negative training samples. While these results show more variation than just the averages shown in Fig. 1, similar conclusions can be

TABLE III: Average Gmean values for OC-SVM+GRKneg and comparative methods

	5 negative samples				10 negative samples				20 negative samples			
	SVM	OC-SVM	variant 7	variant 3	SVM	OC-SVM	variant 7	variant 3	SVM	OC-SVM	variant 7	variant 3
Iris1	**100.0±0.0**	94.5±3.2	97.9±3.1	96.6±3.5	**100.0±0.0**	94.5±3.2	97.9±3.1	97.6±2.9	**100.0±0.0**	94.5±3.2	99.3±1.5	95.9±2.9
Iris2	80.8±9.8	88.9±7.3	**92.7±5.9**	77.4±8.7	**92.3±3.8**	89.8±3.5	88.6±5.9	88.5±3.1	**96.6±2.1**	92.4±3.5	90.8±3.2	92.2±3.5
Iris3	93.2±7.1	89.1±1.5	88.1±18.3	**95.2±5.3**	93.5±4.3	86.8±5.9	94.0±7.1	**94.1±4.8**	**97.0±2.2**	89.1±4.1	94.1±4.8	94.5±6.0
Seed1	77.3±10.3	**86.1±5.6**	84.7±8.4	82.7±7.3	88.0±5.3	85.1±4.5	**91.6±1.2**	86.6±10.2	91.3±3.9	86.1±4.9	**93.1±2.3**	89.5±2.4
Seed2	91.0±4.4	89.7±5.9	**93.9±3.2**	92.5±4.0	**93.5±2.1**	90.0±6.9	92.7±3.7	92.0±2.1	**94.7±1.8**	91.0±3.3	92.2±2.7	92.2±3.3
Seed3	87.4±9.6	91.1±2.2	**92.8±1.7**	90.4±4.0	93.3±1.7	91.9±2.9	**94.0±2.3**	92.8±2.4	**94.9±1.0**	91.6±3.1	92.1±3.2	93.3±3.9
Ion1	48.1±28.9	84.4±8.2	**85.4±8.4**	70.5±9.1	72.8±9.0	**87.2±4.0**	83.3±6.3	79.1±5.2	81.8±5.6	87.9±2.7	88.7±2.2	**89.4±4.9**
Ion2	**65.9±13.8**	25.3±14.4	65.7±19.3	64.0±11.4	**75.4±1.6**	32.4±3.1	72.4±7.9	66.3±11.6	**84.4±6.1**	32.4±3.1	79.3±4.6	77.4±5.2
Son1	48.3±20.8	52.8±4.2	**62.0±10.6**	61.7±6.6	59.4±10.4	54.2±6.2	**69.6±2.2**	55.7±17.9	**73.3±5.6**	53.2±4.5	72.4±6.4	73.1±4.8
Son2	39.0±24.2	57.7±11.0	**62.7±14.8**	42.5±9.0	62.2±5.8	60.9±7.0	**68.8±4.1**	62.2±10.3	69.4±4.2	64.1±4.8	**71.9±5.0**	64.0±14.9
Bank1	95.3±3.0	**96.7±1.6**	91.6±4.3	82.9±10.4	94.3±3.9	**96.7±1.6**	95.9±3.3	93.7±2.6	94.7±2.7	**96.7±1.6**	96.0±1.9	96.2±2.2
Bank2	91.6±9.6	93.3±4.9	93.0±4.8	**93.7±1.5**	**96.0±4.2**	93.3±4.5	93.6±4.7	95.9±2.9	**98.5±1.9**	93.4±2.7	95.3±2.9	95.0±3.0
Happ1	40.6±12.7	34.2±6.8	**52.8±10.6**	46.5±12.8	27.7±17.7	36.2±5.2	48.4±9.0	**52.9±8.0**	51.0±4.3	38.1±5.7	**57.0±5.8**	56.1±9.6
Happ2	21.6±23.1	48.2±8.6	**49.9±11.7**	48.2±9.6	20.7±12.7	51.1±8.0	**56.4±3.9**	51.9±10.3	49.6±3.7	**51.7±9.2**	50.0±10.5	49.8±12.8
Aver.	70.0±12.7	73.7±6.1	**79.5±9.0**	74.6±7.4	76.4±5.9	75.0±4.7	**81.9±4.6**	79.2±6.7	**84.1±3.2**	75.9±4.0	83.7±4.1	82.8±5.7

still drawn. For 5 negative training samples, GRKneg variant 7 achieves the best results in most tasks, while with 20 negative training samples binary SVM already outperforms it in most tasks.

V. Conclusions

This paper proposed a novel approach for small-scale one-class classification with few negative training samples. The proposed approach uses OC-SVM with GRK having negative training and generated samples as reference vectors. The proposed methods clearly outperform standard OC-SVM and also binary SVM for lowest numbers of negative training samples. Wider comparisons of the proposed method with different comparative approaches, such as imbalanced and class-specific classification techniques, remains to be done as future work.

References

[1] Naeem Seliya, Azadeh Abdollah Zadeh, and Taghi M Khoshgoftaar, "A literature review on one-class classification and its potential applications in big data," *Journal of Big Data*, vol. 8, pp. 1–31, 2021.

[2] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft, "Deep one-class classification," in *International conference on machine learning*. PMLR, 2018, pp. 4393–4402.

[3] B. Schölkopf, R.C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," *Advances in neural information processing systems*, vol. 12, 1999.

[4] D.M.J. Tax and R.P.W. Duin, "Support vector data description," *Machine learning*, vol. 54, no. 1, 2004.

[5] Yong-Ping Zhao, Gong Huang, Qian-Kun Hu, and Bing Li, "An improved weighted one class support vector machine for turboshaft engine fault detection," *Engineering Applications of Artificial Intelligence*, vol. 94, pp. 103796, 2020.

[6] Jingxuan Pang, Xiaokun Pu, and Chunguang Li, "A hybrid algorithm incorporating vector quantization and one-class support vector machine for industrial anomaly detection," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 8786–8796, 2022.

[7] F. Sohrab, A. Iosifidis, M. Gabbouj, and J. Raitoharju, "Graph-embedded subspace support vector data description," *Pattern Recognition*, vol. 133, 2023.

[8] Yanshan Xiao, Guitao Pan, Bo Liu, Liang Zhao, Xiangjun Kong, and Zhifeng Hao, "Privileged multi-view one-class support vector machine," *Neurocomputing*, vol. 572, pp. 127186, 2024.

[9] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich, "Deep anomaly detection with outlier exposure," in *International Conference on Learning Representations*, 2019.

[10] Sachin Goyal, Aditi Raghunathan, Moksh Jain, Harsha Vardhan Simhadri, and Prateek Jain, "Drocc: Deep robust one-class classification," in *International conference on machine learning*. PMLR, 2020, pp. 3711–3721.

[11] Yuchun Tang, Yan-Qing Zhang, Nitesh V Chawla, and Sven Krasser, "Svms modeling for highly imbalanced classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 1, pp. 281–288, 2008.

[12] Phuoc Nguyen and Dat Tran, "Repulsive-svdd classification," in *Advances in Knowledge Discovery and Data Mining*. Springer, 2015, pp. 277–288.

[13] Songfeng Zheng, "Smoothly approximated support vector domain description," *Pattern Recognition*, vol. 49, pp. 55–64, 2016.

[14] Zhigang Wang, Zengshun Zhao, Shifeng Weng, and Changshui Zhang, "Solving one-class problem with outlier examples by svm," *Neurocomputing*, vol. 149, pp. 100–105, 2015.

[15] Mikhail Suvorov, Sergey Ivliev, Garegin Markarian, Denis Kolev, Dmitry Zvikhachevskiy, and Plamen Angelov, "Osa: One-class recursive svm algorithm with negative samples for fault detection," in *Artificial Neural Networks and Machine Learning*. Springer, 2013, pp. 194–207.

[16] Van Nguyen, Trung Le, Thien Pham, Mi Dinh, and Thai Hoang Le, "Kernel-based semi-supervised learning for novelty detection," in *2014 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2014, pp. 4129–4136.

[17] Yaohui Ding and Arun Ross, "An ensemble of one-class svms for fingerprint spoof detection across different fabrication materials," in *EEE International Workshop on Information Forensics and Security*. IEEE, 2016, pp. 1–6.

[18] Suyu Mei and Hao Zhu, "A novel one-class svm based negative data sampling method for reconstructing proteome-wide htlv-human protein interaction networks," *Scientific reports*, vol. 5, no. 1, pp. 8034, 2015.

[19] Dieter Oosterlinck, Dries F Benoit, and Philippe Baecke, "From one-class to two-class classification by incorporating expert knowledge: Novelty detection in human behaviour," *European Journal of Operational Research*, vol. 282, no. 3, pp. 1011–1024, 2020.

[20] J. Raitoharju and A. Iosifidis, "Generalized reference kernel for one-class classification," in *International Joint Conference on Neural Networks*, 2022.

[21] P. Drineas and M.W. Mahoney, "On the Nystrom Method for Approximating a Gram Matrix for Improved Kernel-based Learning," *Journal of Machine Learning Research*, vol. 6, pp. 2153–2275, 2005.

[22] Alex Gittens and Michael W Mahoney, "Revisiting the nyström method for improved large-scale machine learning," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 3977–4041, 2016.

[23] Nir Ailon and Bernard Chazelle, "The fast johnson–lindenstrauss transform and approximate nearest neighbors," *SIAM Journal on Computing*, vol. 39, no. 1, pp. 302–322, 2009.

[24] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," *Neural Information Processing Systems*, 2007.

[25] Zhenyu Liao, Romain Couillet, and Michael W Mahoney, "A random matrix analysis of random fourier features: beyond the gaussian kernel, a precise phase transition, and the corresponding double descent," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13939–13950, 2020.