OCFSP: Self-supervised One-class Classification Approach Using Feature-slide Prediction Subtask for Feature Data

Toshitaka Hayashi
University of Hradec Kralove: Univerzita Hradec Kralove
Hamido Fujita (HFujita-799@acm.org)
Iwate Prefectural University https://orcid.org/0000-0001-5256-210X

Research Article

Keywords: Machine Learning, One-class Classification, Self-supervision

Posted Date: February 21st, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1212211/v1

OCFSP: Self-supervised One-class Classification Approach using Feature-slide Prediction Subtask for Feature Data

Toshitaka Hayashi[1] [0000-0002-7599-4404], Hamido Fujita [2, 3,4,*] [0000-0001-5256-210X]

[1] Faculty of Science, University of Hradec Kralove, Hradec Kralove, Czech Republic
toshitaka.hayashi@uhk.cz
[2] Faculty of Information Technology, HUTECH University, Ho Chi Minh City, Vietnam
[3] Regional Research Center, Iwate Prefectural University, Takizawa, Japan
[4] i-somet Inc., Morioka, Japan
h.fujita@hutech.edu.vn, hfujita-799@acm.org

Abstract. One-class classification (OCC) is a machine learning problem where training data has only one class. Recently, self-supervised OCC algorithms have been increasing attention. These algorithms train the model for pretext tasks and use the model error for OCC. However, these tasks are specialized for images, and applying them to feature data is not practical or appropriate for such purpose. This paper proposes a one-class classification approach using feature-slide prediction (FSP) subtask for feature data (OCFSP). In particular, the self-labeled dataset is created from training data. In which additional feature vectors are generated by sliding original vectors and self-annotated as the number of the feature slide. Such a dataset is used to train a multi-class classifier, which aims to predict the number of the feature slides. Since this classification model is built using data from only one class, the FSP accuracy for seen data is high relative to unseen data. Accordingly, OCC could be made using the accuracy of FSP. Proposed methods are experimented with using the imbalanced-learn dataset.

Keywords: Machine Learning, One-class Classification, Self-supervision.

1 Introduction

In recent years, Machine Learning (ML) has been introduced to various fields. Especially, supervised learning is widely applied with annotation by experts [1, 2]. However, such methods require a large volume of data. Moreover, model accuracy worsens due to the dataset problems, such as data imbalance [3] and outlier [4].

Besides, the supervised learning model cannot predict classes that are not in training data. Especially, there is a case that only one class is collectible as training da-ta. Since all class labels are the same in training data, the supervised model classifies all data as the same class. This problem is called one-class classification (OCC) [5], which is an important issue in ML and related to anomaly detection [6], novelty detection [7]. Intrusion detection [8], and zero-shot learning [9]. The objective of OCC is classifying input data into a seen class or the rest of unseen classes. These classes are defined as included in training data or not, respectively.

For this purpose, various OCC algorithms are proposed. Early studies are shallow methods, such as OCSVM [10], Local Outlier Factor (LOF) [11], and Isolation Forest (IF) [12]. These methods are effective for feature data. However, shallow methods have limitations for image data since there is no feature extraction process, such as the convolutional layer [13].

Recently, DL-based OCC methods have been proposed for image datasets [13]. These methods are roughly classified into three groups, shallow methods with feature extraction [13], fake-unseen-samples approach [14], and self-supervised approach [15, 16]. In which, the last approach is the best in terms of accuracy [15, 16]. These methods consider a pretext task for data. Then, the model for such a task is trained using training data. Since all training data is seen, model error for seen data is small relative to unseen data. Therefore, OCC could be made using model error for the pretext task.

The motivation of this study is to apply the self-supervised approach to feature data. Self-supervised OCC algorithms show the best accuracy in the image datasets. However, these pretext tasks are specialized to image [15, 16], and applying these tasks to vector data is not practical or suitable for such a purpose. Accordingly, considering effective subtask for feature data is a significant challenge.

Existing pretext tasks are roughly classified into two groups, generation (regression) [16] or classification [15]. Generation tasks are related to reconstruction or transformation, and these methods have an advantage in terms of processing speed. In contrast, classification tasks aim to classify the self-labeled dataset created according to defined classification subtask, and these tasks have an advantage for accuracy. These aspects are trade-off practices [16].

This study prioritizes accuracy and selects classification tasks. Then, the main problem is what kind of classification should be applied. In image data, pretext tasks are such as rotation classification [17], classification of geometric transformation [15], and image perturb classification [18]. However, these tasks are not possible in feature data because rotation or geometric transformations are not applicable.

Accordingly, One-class classification approach using feature slide prediction (OCFSP) is proposed in the conference paper [19]. In which, a novel subtask, namely feature-slide prediction (FSP), is proposed as a pretext task for feature data. This task uses a self-labeled dataset, which includes additional feature vectors created by sliding dimensions of original vectors. These additional vectors are annotated as to how features are slid. Then, a multi-class classifier is trained for classifying these feature slides. Since this classification model is built using data from only one class, the accuracy of FSP for seen data is high relative to unseen data. Therefore, seen and unseen classes could be discriminated based on the accuracy of FSP.

This paper is an extension of such a conference paper. In which FSP is made with further classification algorithms. Moreover, OCFSP is compared with other OCC algorithms in multiple train test split. In which OCFSP has relatively high accuracy where training data is small size.

The contributions of this study are listed as follows:

- Novel One-class classification algorithm, namely OCFSP, is proposed. The main originality is the FSP subtask. In which the self-labeled dataset is created by sliding feature vector. Then, the classification model is trained to predict the number of feature slides. To our knowledge, this is the first classification subtask for feature data.
- OCFSP is experimented with using the imbalanced-learn dataset. Moreover, OCFSP is compared with other OCC algorithms. OCFSP shows a high AUC score where seen data is in the narrow distribution. Moreover, OCFSP outperforms other methods where training data is small size.
- Time complexity is analyzed. OCFSP shows fast processing speed in the testing stage, which is different from classification subtasks in image datasets. Such difference is related to the memory size of data.

The organization of the paper is summarized as follows. Section 2 describes related work, such as OCC and self-supervised OCC. Section 3 presents the proposed OCFSP framework. Section 4 and section 5 provide experiment results and discussions. Finally, section 6 gives the conclusion and future work.

2 Related Work

2.1 One-class classification

OCC is a promising research area because it can detect unseen samples, which is the weak point of supervised learning. In this problem, only one class is seen as training data, and other classes are unseen. The main challenge is how to detect unseen data without training.

In addition, combining multiple one-class classifiers is a promising solution for multi-class training data. In which, one-class classifiers are trained in class by class. Then, testing samples are predicted based on all classifiers in the manner of ensemble learning [20]. Such kind of strategy is called one-class ensemble [21-23], which is promising research in detecting unseen samples. Moreover, one-class ensemble is effective for data imbalance problem. Since the models are trained from one class, data balance is not a problem [23]. Such an approach is applicable in many situations.

Fig. 1 shows general OCC framework, which has two stages, training, and testing. In the training stage, the behavior of a seen class is determined based on some algorithms. Then, classification is made by whether testing data includes seen behavior.

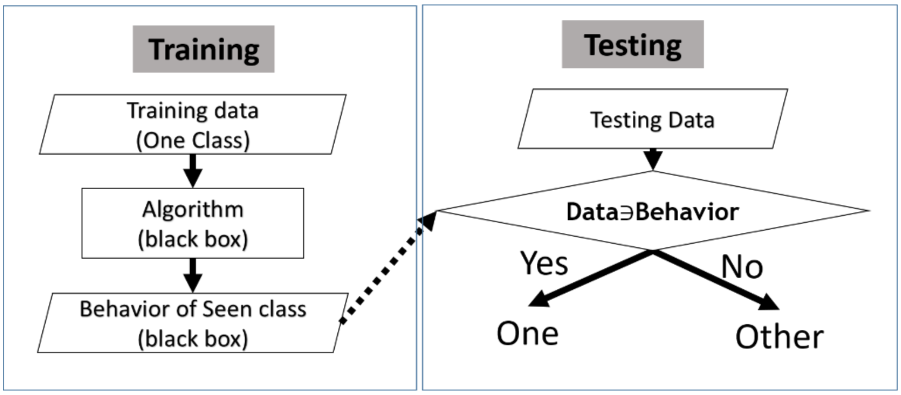

Fig. 1. One-class classification framework

For this purpose, several OCC algorithms are proposed. One-class Support Vector Machine (OCSVM) applies mapping function for seen data into feature vector space. In such vector space, unseen samples are considered origin O. Then, the maximum margin hyper-plane between mapped seen vectors and O is computed [10]. In contrast, Local Outlier Factor (LOF) computes outlier scores of a sample using local density of original sample and the neighbor samples [11]. Such kind of scoring becomes large where data is far from the neighbor samples [11]. Additionally, Isolation Forest (IF) is the technique to detect outlier using tree structure with random split. In such tree, outlier is isolated with high probability since it is far from normal samples [12]. Further-more, recent studies extend these algorithms [4, 21, 24]. In addition, cluster-based method generates clusters from seen class. Then, data which is not assigned to cluster are considered as unseen [25].

Apart from these studies, DL-based OCC are developed for image data [13]. These methods are roughly classified into three groups, feature extraction + shallow method [13], fake-unseen approach [14], and self-supervised OCC [15,16]. The first method is extension of shallow method. In contrast, the second approach generates fake-unseen-samples and apply supervised classification [14]. However, such approach is difficult since there is no information for unseen samples. In other hand, self-supervised OCC considers any subtask and trains ML model for such subtask. Such methods are the best in terms of accuracy [15,16].

2.2 Self-supervised One-class Classification

Recently, self-supervised OCC is a promising framework and increasing attention. Such a framework considers pretext tasks for training data. Then, the ML model is trained for such subtask. Since training data has only seen class, model error for seen data is small relative to unseen data. Therefore, OCC could be made using model error of pretext task.

Several pretext tasks are proposed for self-supervised OCC [15, 16, 18, 26, 27]. These tasks are roughly classified into two groups, generation, and classification. Generative tasks, such as reconstruction and transformation, have an advantage in terms of processing speed [16]. However, there is a limitation for accuracy.

In contrast, classification tasks aim to classify the self-labeled dataset. Such a dataset includes original data and additional data (that is created using original data). In which self-labels correspond to how data are created. These tasks have an advantage in terms of accuracy [15]. However, this approach takes time to create the self-labeled dataset. These aspects are trade-off practices.

Self-supervised OCC is mainly used for image data. In which various subtasks are proposed, such as classification of geometric transformation [15], perturb classification [18], and image transformation to one image [16]. However, these subtasks are specialized for image data. Therefore, applying to other data types is not suitable for such a purpose.

Self-supervised techniques have been applied to time-series data in the context of anomaly detection [26, 27]. Baldacci et al. use the gas consumption forecasting task for time-series anomaly detection [26]. Blázquez-García et al. proposed classification sub-task for water leak detection [27]. In which, a self-labeled dataset is created with additional signals generated by multiplications of original signals [27].

On the other hand, this paper proposes a classification subtask for feature data, namely FSP subtask. In which the self-labeled dataset is created with sliding original feature vectors. To the best of our knowledge, FSP is the first classification subtask for feature data.

3 One-class Classification using feature-slide prediction subtask

In this section, novel one-class classification algorithm OCFSP is presented. In particular, FSP is proposed as a new subtask for feature data. **Fig. 2** shows the framework, which consists of two stages, training, and testing. In which, only seen data is used as training data.

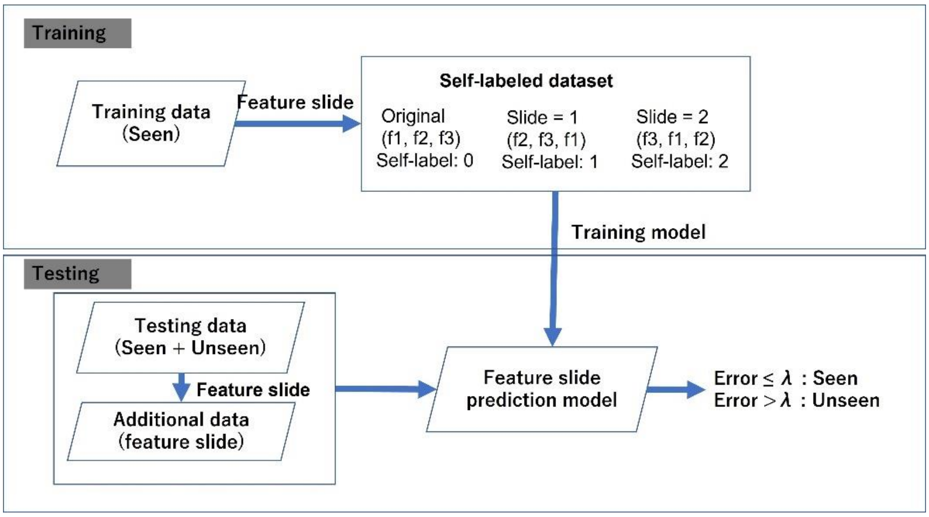

Fig. 2. OCFSP framework

In the training stage, additional data are generated by sliding feature vectors. Then, self-labeled dataset is created by gathering original and these additional data. In which, self-label represents how many slides are applied. Then after, classification model is trained using such a created dataset. Finally, threshold value is computed based on accuracy of FSP model.

In the testing stage, additional data is generated in the same way as training. Then, FSP model is applied, and the accuracy is computed. If accuracy is higher than a threshold value, data is treated as seen class. Otherwise, data is concluded as unseen class. In the following paragraphs, mathematical descriptions are provided.

Data is defined as d-dimensional feature vector X, as shown in equation (1).

$$X = (X_1, X_2, \dots, X_d) \tag{1}$$

Where d is number of dimensions. The objective in OCC is predicting class label Y that is defined as in equation (2).

$$Y = \{S, U\} \tag{2}$$

Where S and U are seen and unseen class, respectively.

This study proposes FSP subtask. Such task generates additional data A. In which, feature slide T is computed to generate additional data A from X as equation (3).

$$\begin{gathered} T(X, z): X \rightarrow A \\ A = \{A_1, \dots, A_z\} \end{gathered} \tag{3}$$

In which, z is number of applied slides where 0 < z < d because d-dimensional data has d-1 possible slides. Moreover, volume of A increases related to z. Additionally, the components of A are defined as equation (4).

$$\begin{gathered} A_1 = (X_2, X_3, \dots, X_d, X_1) \\ \vdots \\ A_z = (X_{z+1}, \dots, X_d, X_1, \dots, X_z) \\ \vdots \\ A_{d-1} = (X_d, X_1, \dots, X_{d-1}) \end{gathered} \tag{4}$$

In which, original feature vector X is slid forward by the number of the slide. These original and additional data are annotated using self-label *SL* as in equation (5).

$$SL = \{0,1, \dots, z\} \tag{5}$$

In which, original data is self-labeled as 0. In contrast, additional data are labeled as number of the slides. Accordingly, FSP model g is defined as following equation (6).

$$g: X \,, A \to SL \tag{6}$$

In which, g aims to predict SL which is number of feature slides.
Finally, score of data related to seen class is computed using original data *X* and additional data A as in equation (7).

$$\text{Score}(X) = L(\text{g}(X) = 0 \mid X) + \textstyle\sum_{k=1}^{z} L(g(A_k) = k \mid A_k) \tag{7}$$

Where k is the value of each slide. This score is computed using likelihood function related to correct prediction. Likelihood provides a more detailed score value than general accuracy, where data size is small. Therefore, such kind of scoring highlights the difference between seen and unseen classes.
In the following sub-sections, training stage and testing stage are presented.

3.1 Training stage

In the training stage, dataset *Dtr* is defined as equation (8):

$$Dtr = [Xtr_1, Xtr_2, \dots, Xtr_N] \tag{8}$$

Where N is number of the data. Besides, additional datasets are created by sliding *Dtr* as in equation (4). These datasets are defined as in equation (9):

$$\begin{gathered} Atr_1 = [Atr_{1,1}, Atr_{1,2}, \dots, Atr_{1,N}] \\ \vdots \\ Atr_z = [Atr_{z,1}, Atr_{z,2}, \dots, Atr_{z,N}] \end{gathered} \tag{9}$$

Besides, Self-labeled dataset *Dself*; is created by merging *Dtr* and *Atr* as in equation (10):

$$Dself = [Dtr, Atr_1, \dots, Atr_z] \quad (10)$$

Then, self-label *SL* is assigned to these dataset as shown in equation (5). Besides, feature-slide classifier *g* is trained using *Dself* and *SL* as shown in equation (6). In such process, existing classification algorithms are applied. Since training is done using only seen samples, this classifier has high accuracy for seen class relative to unseen class. Therefore, threshold value between seen and unseen, could be computed using accuracy for seen data. Such computation is made in heuristic optimal way.

3.2 Testing stage

In the testing stage, input is *Xtest*, and additional data *Atest* are generated from *Xtest* as shown in equation (4). These data are merged to self-labeled testing set *Dtest* as shown in equation (11):

$$Dtest = [Xtest, Atest_1, \dots, Atest_z] \quad (11)$$

Then, score of *Xtest* is computed as equation (12).

$$\mathrm{score}(Xtest) = L(\mathrm{g}(Xtest) = 0 \mid Xtest) + \sum_{k=1}^{z} L(g(Atest_k) = k \mid Atest_k) \quad (12)$$

Such equation is based on previous equation (7)
Finally, seen-unseen classification *f* is established using equation (13).

$$f(Xtest) = \begin{cases} S\ (score(Xtest) \geq \lambda) \\ U\ (score(Xtest) < \lambda) \end{cases} \quad (13)$$

Where λ is a threshold value, which is determined in a heuristic optimal way.

4 Experiment

The proposed method (OCFSP) has been validated using data listed in section 4.1. The measurement of evaluation is shown in section 4.2. The experiment results are shown in section 4.3.

4.1 The data

OCFSP is evaluated with an imbalanced learn dataset [28], which is implemented as a Python package. Such dataset consists of 27 sub-datasets with class imbalance for binary classification [28].

Table 1 shows information of datasets, such as dimension, number of each class, and Imbalance Ratio (IR). All datasets are normalized based on min-max. In the experiment, one class is treated as seen class, and another class is concluded as unseen class.

Table 1. Details of imbalanced datasets in experiment

Data	Dimension	Minority	Majority	IR
ecoli	7	35	301	8.60
optical_digits	64	554	5066	9.14
satimage	36	626	5809	9.28
pen_digits	16	1055	9937	9.42
abalone	10	391	3786	9.68
sick_euthyroid	42	293	2870	9.80
spectrometer	93	45	486	10.80
car_eval_34	21	134	1594	11.90
isolet	617	600	7197	12.00
us_crime	100	150	1844	12.29
yeast_ml8	103	178	2239	12.58
scene	294	177	2230	12.60
libras_move	90	24	336	14.00
thyroid_sick	52	231	3541	15.33
coil_2000	85	586	9236	15.76
arrhythmia	278	25	427	17.08
solar_flare_m0	32	68	1321	19.43
oil	49	41	896	21.85
car_eval_4	21	65	1663	25.58
wine_quality	11	183	4715	25.77
letter_img	16	734	19266	26.25
yeast_me2	8	51	1433	28.10
webpage	300	981	33799	34.45
ozone_level	72	73	2463	33.74
mammography	6	260	10923	42.01
protein_homo	74	1296	144455	111.46
abalone_19	10	32	4145	129.53

In which, training data and testing data are split by three ratios, such as 80%:20%, 60%:40%, and 10%:90%. Then, the training set is split into minority and Majority data, and the one-class classifiers are trained separately. Each split is applied five times, and these average scores are reported as the experiment result.

4.2 Measurement of the evaluation

Evaluation is done using the Area under the ROC Curve (AUC). This curve is a graph plotting the performance in all possible thresholds. In which the x-axis and y-axis are FPR and TPR, respectively. These values are computed as given in Eq. (14) – Eq. (15) and **Table 2**. In which, Positive and Negative are corresponding to minority and Majority class, respectively.

$$\text{TPR} = \frac{TP}{TP+FN} \tag{14}$$

$$\text{FPR} = \frac{FP}{FP+TN} \tag{15}$$

Table 2. Confusion Matrix

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

4.3 Experiment result

The experiment is made with the following three parts: 1) Considering appropriate number of the slide, 2) Compare with other OCC algorithms, and 3) Compute processing time.

In this experiment, OCC algorithms and supervised classification algorithms are applied using scikit-learn package [29]. **Table 3** provides applied packages for algorithms. Baseline methods are OCC algorithms. In addition, OCFSP uses supervised classification algorithms for FSP model. In which AUC scores are computed using outputs of score functions as shown in **Table 3**. Such kind of scorings are likelihood related to seen class, or correct slide prediction, respectively.

Table 3. Applied packages for the experiment

Method		**Scikit-learn Package**	**Score function**
Baseline OCC	OCSVM	OneClassSVM	score_samples
	LOF	LocalOutlierFactor	score_samples
	IF	IsolationForest	score_samples
	GMM	GaussianMixture	score_samples
OCFSP	DT	DecisionTreeClassifier	predict_proba
	LR	LogisticRegression	predict_log_proba
	GNB	GaussianNB	predict_log_proba
	MLP	MLPClassifier	predict_proba

In addition, FSP is made by four classification algorithms, such as Decision Tree (DT), Logistic Regression (LR), Gaussian Naïve Bayes (GNB) and Multi-layer Perceptron (MLP). These algorithms are selected from algorithms included in the scikit-learn library [29]. In which, processing time is considered for model selection because model training takes time when the size of the self-labeled dataset increases. All algorithms are applied with default parameters.

The first experiment is made for considering appropriate number of the feature slide for OCFSP. **Fig. 3** shows the average AUC score for imbalanced-learn dataset. These results are corresponding to seen classes, train test split, and classification algorithms for FSP. In addition, Z represents number of the slides applied to original data. Therefore, Z+1 class classification is made as pretext task.

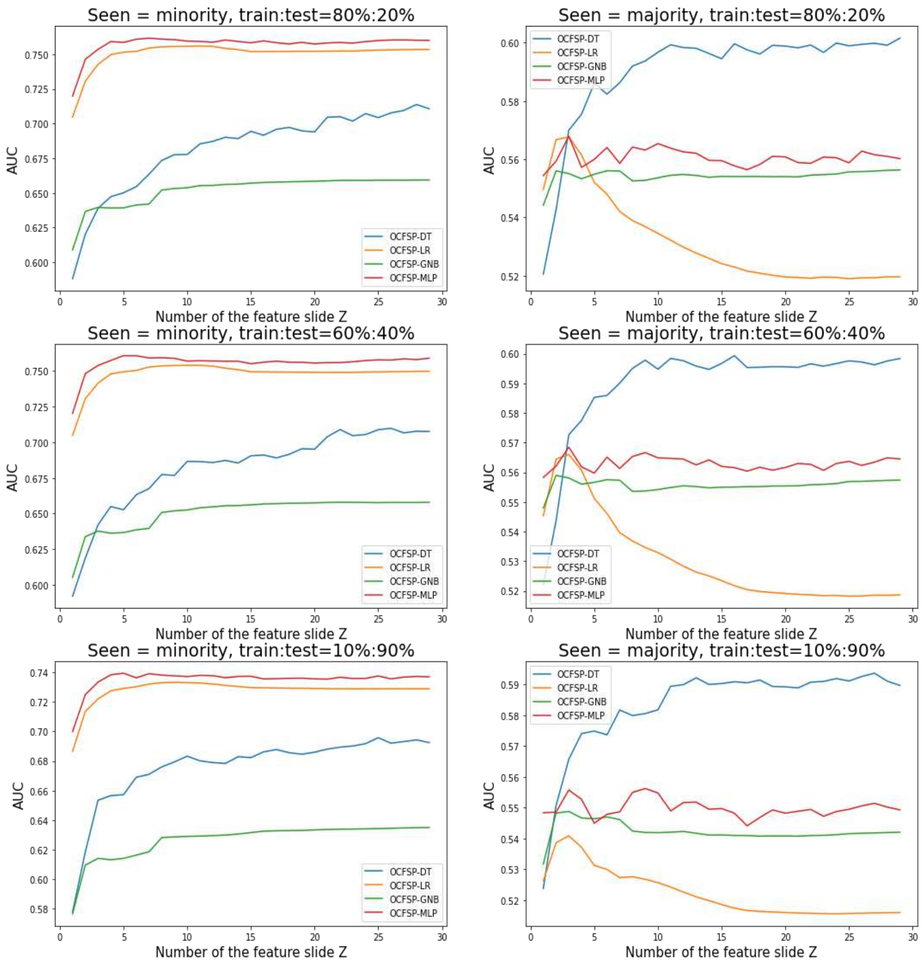

Fig. 3. The average AUC score for OCFSP

In such figure, AUC is small, where the z value is small. The reason is considered as random prediction leads to high accuracy where number of the class is small. In such a case, FSP for unseen class becomes unfairly high accuracy. On the other hand, the AUC score decreases where the z value is large. Such problem is due to FSP becomes unpredictable for even seen class.

Besides, **Table 4** shows the best AUC scores for each pair of seen class and classification algorithm. Overall, MLP is the best classifier where seen class is minority. In contrast, DT shows the best performance for majority seen class. In addition, larger training data provides higher AUC score.

Table 4. The best AUC score and number of the feature slide Z

	Seen = minority			Seen = majority		
	80%:20%	60%:40%	10%:90%	80%:20%	60%:40%	10%:90%
OCFSP-DT	71.4 z=28	71.0 z=26	69.6 z=25	**60.2** z=28	**59.9** z=16	**59.4** z=27
OCFSP-LR	75.6 z=11	75.4 z=11	73.3 z=9	56.8 z=3	56.4 z=3	54.1 z=3
OCFSP-GNB	65.9 z=29	65.8 z=22	63.5 z=29	55.6 z=29	55.9 z=2	54.9 z=3
OCFSP-MLP	**76.2** z=7	**76.1** z=5	**73.9** z=5	56.8 z=3	56.8 z=3	55.6 z=9

In the following experiments, number of the feature slide z are selected as shown in **Table 4**. In some datasets, z is larger than the dimension of the data. In such a case, the reported result is AUC score where $z = d - 1$.

Besides, the performance for the minority classifier is better than the majority classifier. The reason is considered the seen data distribution. Perhaps, the minority class has a narrower distribution than the majority class. In which the distribution of the feature slide is narrow in the same way.

On the other hand, the majority class and such feature slide vectors should have wide distribution. In such a case, self-labeled data distribution should overlap, and feature slide prediction becomes difficult. In which model training is not successful, and such model error cannot discriminate between seen and unseen classes.

4.4 Comparison with other OCC algorithms

The proposed OCFSP is compared with other OCC algorithms. Comparison is made with four OCC algorithms. These methods are applied with default parameters of scikit-learn package [29].

- One-class Support Vector Machine (OCSVM) [10] uses mapping function from seen data into feature vector space. In such space, the maximum margin hyperplane is computed between mapped seen data and Origin O that is regarded as unseen.
- Local Outlier Factor (LOF) [11] compute outlier scores for sample. Such score is calculated with density of original samples and these neighbor samples. Outlier score is large where data is far from the neighbor samples.
- Isolation Forest (IF) [12] uses tree structure with random split. Such tree is regarded to assign seen data into the same place and isolate unseen data. Therefore, score could be computed from where data is assigned.
- Gaussian Mixture Model (GMM) [30] is known as unsupervised clustering algorithm, which could be applicable to discriminate seen and unseen data. As a process, GMM is applied for training data and clusters are generated. Then, score related to these cluster is computed for each data. Since clusters are created from seen class, score is related to seen class. Therefore, data with small score is concluded as unseen class [25].

Tables 5-7 shows the comparison where seen data is minority class. In particular, **Table 5** provides the result where train test split is 8:2. In which, OCFSP methods show the best AUC scores for 10 datasets in total. In addition, OCFSP-MLP is tie with baseline methods in terms of average AUC score.

Table 5. Comparison where seen data is minority, train test split is 8:2.

Dataset	Baseline OCC				OCFSP			
	OCSVM	LOF	IF	GMM	DT	LR	GNB	MLP
ecoli	89.0	89.4	**91.8**	84.1	50.6	86.6	77.4	90.7
optical_digits	97.4	**98.8**	95.3	97.2	94.4	86.9	82.6	83.9
satimage	90.6	83.2	**91.8**	79.0	86.6	74.3	81.1	73.7
pen_digits	99.0	**99.8**	98.9	99.4	99.3	84.0	83.1	97.8
abalone	80.5	75.7	**83.8**	82.7	59.9	64.4	26.5	62.5
sick_euthyroid	70.6	77.4	77.6	**84.7**	65.8	77.4	57.4	76.9
spectrometer	48.5	26.3	56.4	49.6	**69.3**	63.4	52.0	55.3
car_eval_34	**99.9**	98.2	59.4	99.8	95.0	98.7	98.5	98.5
isolet	93.1	**96.2**	92.3	93.7	72.1	78.7	64.6	81.4
us_crime	72.0	71.8	66.7	69.3	72.4	81.1	71.0	**82.6**
yeast_ml8	**58.7**	58.5	56.0	45.3	56.3	52.3	50.3	52.6
scene	62.7	59.7	63.0	59.0	**64.7**	63.5	62.7	60.6
libras_move	92.4	67.9	91.6	91.6	82.7	93.6	78.7	**95.7**
thyroid_sick	71.3	60.8	74.5	77.5	64.7	**77.9**	56.2	77.6
coil_2000	55.3	52.9	56.3	54.6	56.1	66.0	52.0	**66.9**
arrhythmia	58.4	58.2	61.9	63.9	**71.2**	68.3	50.0	66.9
solar_flare_m0	**61.0**	52.5	52.7	56.9	55.2	58.9	60.0	57.9
oil	68.2	56.7	49.8	**78.4**	64.3	52.0	50.6	52.7

car_eval_4	**99.9**	97.9	24.9	99.8	99.0	99.4	97.9	99.4
wine_quality	47.3	45.7	50.2	46.9	**62.8**	58.6	56.7	59.7
letter_img	89.9	**98.4**	89.8	97.5	97.6	90.8	83.5	93.8
yeast_me2	83.0	85.9	77.3	70.0	52.1	87.7	51.6	**88.8**
webpage	62.1	**84.0**	52.9	79.1	69.6	70.9	69.8	71.0
ozone_level	76.7	78.7	**81.6**	67.2	69.8	80.3	58.1	79.6
mammography	79.4	74.5	80.1	79.8	69.2	81.6	**88.5**	85.1
protein_homo	**80.1**	75.2	79.8	79.1	60.7	78.4	50.0	78.7
abalone_19	70.9	61.2	**73.2**	71.6	65.7	65.3	69.3	66.0
Average	**76.2**	73.5	71.5	**76.2**	71.4	75.6	65.9	**76.2**

Table 6 shows the comparison where train test split is 60%:40%. In which, OCFSP methods show the best scores for 11 datasets in total. In addition, OCFSP-MLP is the 0.1 point behind in the average score, which is comparable with the baseline.

Table 6. Comparison where seen data is minority, train test split is 6:4.

Dataset	Baseline OCC				OCFSP			
	OCSVM	LOF	IF	GMM	DT	LR	GNB	MLP
ecoli	91.5	58.1	**92.1**	87.8	53.9	88.3	76.2	92.0
optical_digits	97.6	**98.8**	95.3	97.2	94.1	87.0	83.8	85.0
satimage	91.5	83.7	**92.4**	79.9	86.7	74.7	81.2	74.5
pen_digits	99.0	**99.8**	98.9	99.5	99.3	83.4	83.0	97.4
abalone	81.8	70.2	**84.6**	82.8	62.8	66.9	27.0	58.6
sick_euthyroid	69.5	73.0	75.4	**83.5**	67.0	76.3	57.4	74.9
spectrometer	52.7	32.2	59.2	52.4	**72.9**	63.6	52.7	59.8
car_eval_34	**99.8**	96.6	63.5	99.6	94.6	98.6	98.4	98.6
isolet	93.0	**96.3**	92.2	94.9	77.2	79.1	64.4	80.9
us_crime	73.4	71.5	69.5	68.8	71.2	79.8	70.8	81.5
yeast_ml8	56.3	**57.9**	57.0	47.4	52.2	50.5	50.3	51.7
scene	62.6	59.7	63.5	58.7	**63.8**	**63.8**	60.6	60.6
libras_move	90.5	63.3	85.3	88.4	81.2	88.5	75.5	**94.0**
thyroid_sick	71.7	62.4	76.1	78.3	64.2	**80.4**	56.0	80.2
coil_2000	54.5	52.9	55.9	54.1	56.4	65.7	52.0	**66.3**
arrhythmia	63.1	61.9	59.5	65.6	**67.9**	66.3	50.0	63.4
solar_flare_m0	56.7	51.0	49.5	56.2	48.5	57.6	**60.6**	59.4
oil	63.6	52.1	49.1	**66.7**	53.4	53.6	50.4	53.1
car_eval_4	**99.9**	95.7	36.8	99.8	99.2	99.4	97.8	99.3
wine_quality	49.3	47.3	50.9	47.9	**61.6**	58.9	57.4	60.0
letter_img	89.7	**97.5**	88.2	97.4	97.5	90.9	84.0	94.1
yeast_me2	79.4	82.9	77.3	73.5	53.0	83.3	50.7	**84.9**
webpage	62.2	79.2	53.9	**80.1**	69.2	70.9	69.5	72.0

ozone_level	78.5	80.4	**83.4**	73.0	67.1	81.3	58.5	81.1
mammography	79.7	81.5	80.3	79.3	73.2	81.8	**89.8**	84.7
protein_homo	**80.8**	75.1	80.5	79.6	62.0	78.0	50.0	78.1
abalone_19	69.7	59.5	**73.6**	64.0	66.3	67.0	68.3	67.7
Average	**76.2**	71.9	72.0	**76.2**	71.0	75.4	65.8	76.1

In the same way, **Table 7** provides the comparison where train test split is 1:9. OCFSP outperform baseline in 12 datasets. In addition, OCFSP-MLP shows the best AUC score in the compared methods.

Table 7. Comparison where seen data is minority, train test split is 1:9.

Dataset	Baseline OCC				OCFSP			
	OCSVM	LOF	IF	GMM	DT	LR	GNB	MLP
ecoli	**91.0**	88.4	66.3	89.3	78.1	90.0	50.1	90.0
optical_digits	97.0	**97.2**	94.3	89.4	87.6	86.8	81.5	84.4
satimage	91.3	90.2	**92.2**	77.1	85.6	76.1	80.8	77.5
pen_digits	**98.8**	96.0	98.0	99.2	**98.8**	79.3	82.0	96.5
abalone	81.0	72.2	**83.9**	81.9	68.7	67.6	26.3	64.8
sick_euthyroid	70.5	69.3	67.8	8**1.9**	75.5	79.6	55.5	76.9
spectrometer	59.3	40.4	44.6	**66.9**	56.0	56.0	53.1	57.2
car_eval_34	95.8	75.3	61.2	**97.3**	85.9	94.4	96.4	94.6
isolet	92.8	92.1	90.5	**96.0**	64.9	79.2	65.8	83.5
us_crime	76.8	66.8	67.8	70.1	70.7	85.1	71.1	**85.3**
yeast_ml8	**55.8**	54.8	54.0	51.3	53.2	50.9	50.9	51.4
scene	61.0	59.4	59.1	57.9	57.7	**63.8**	54.8	58.6
libras_move	59.3	45.3	50.0	58.7	**64.3**	61.3	62.2	58.4
thyroid_sick	73.5	68.3	75.3	78.1	72.9	**80.9**	56.9	80.4
coil_2000	54.1	54.0	56.8	50.7	54.6	64.7	50.9	**65.7**
arrhythmia	57.2	57.4	50.0	57.4	55.1	**59.3**	50.0	58.1
solar_flare_m0	59.6	49.8	33.0	60.9	55.7	58.3	**64.8**	61.0
oil	**60.8**	50.7	48.0	59.9	51.6	51.9	49.6	52.5
car_eval_4	**98.1**	79.9	53.5	97.9	94.9	97.2	96.7	97.2
wine_quality	49.9	49.5	53.1	51.1	57.7	55.9	**58.0**	56.2
letter_img	89.4	89.3	84.2	**96.3**	94.6	89.3	83.3	91.8
yeast_me2	61.9	61.0	66.0	58.3	54.9	**84.4**	50.1	84.0
webpage	61.6	60.4	50.0	**76.5**	63.9	69.5	70.1	72.3
ozone_level	73.2	62.1	79.8	79.5	69.6	80.8	55.5	**81.0**
mammography	80.8	76.7	81.1	80.0	77.2	82.3	**88.7**	83.3
protein_homo	**81.4**	70.8	79.8	77.8	67.8	73.3	50.0	74.0

abalone_19	56.2	59.1	**62.3**	54.4	61.2	61.6	59.2	59.7
Average	73.6	68.0	66.8	**73.9**	69.6	73.3	63.5	**73.9**

Besides, **Tables 8-10** show the experiment result where majority class is the seen class. In particular, **Table 8** provides comparison where train test split is 8:2. In which OCFSP shows the best scores in four datasets. On the other hand, OCFSP underperforms baseline in the average score.

Table 8. Comparison where seen class is majority, split is 8:2

Dataset	Baseline OCC				OCFSP			
	OCSVM	LOF	IF	GMM	DT	LR	GNB	MLP
ecoli	80.5	65.7	78.4	**81.6**	50.2	28.3	41.0	36.6
optical_digits	21.2	**97.7**	44.9	75.6	75.1	52.8	48.5	51.9
satimage	23.8	51.6	38.8	40.3	64.2	50.1	55.4	**71.0**
pen_digits	78.8	**98.3**	81.8	83.9	87.7	76.9	64.3	67.9
abalone	52.4	43.9	59.8	31.8	49.9	70.9	**79.8**	69.7
sick_euthyroid	42.8	**61.1**	42.1	43.7	48.5	37.0	42.9	46.6
spectrometer	83.6	**91.9**	88.4	72.5	65.7	49.7	49.3	56.6
car_eval_34	**98.6**	50.0	90.4	98.3	58.5	93.0	96.0	82.1
isolet	22.9	**77.5**	26.2	37.1	48.0	35.5	47.3	37.2
us_crime	79.5	73.7	**82.0**	75.6	71.3	76.8	65.5	74.0
yeast_ml8	48.6	47.6	49.6	**60.2**	54.3	49.1	52.6	48.0
scene	49.2	**52.8**	47.4	47.3	52.0	44.7	47.1	46.5
libras_move	82.1	91.3	73.5	95.6	87.7	92.4	53.7	**96.4**
thyroid_sick	52.4	**68.7**	49.0	46.1	50.2	61.7	47.5	62.1
coil_2000	53.9	**56.0**	54.1	53.4	51.5	38.5	50.3	37.0
arrhythmia	46.3	41.3	46.0	40.2	45.8	35.9	**48.6**	35.8
solar_flare_m0	76.7	61.7	**77.7**	77.4	65.7	62.6	70.0	63.8
oil	60.3	75.6	73.8	**82.6**	59.4	57.4	49.7	58.7
car_eval_4	**99.2**	50.0	89.0	97.4	55.5	86.7	94.8	76.0
wine_quality	66.8	69.5	70.6	**72.0**	65.5	55.8	62.7	58.7
letter_img	59.6	**93.9**	51.5	52.6	67.8	62.4	47.1	69.0
yeast_me2	**78.2**	67.4	74.6	73.3	49.6	36.4	25.3	31.8
webpage	49.5	72.7	40.1	**89.7**	82.7	77.6	88.1	76.9
ozone_level	**55.1**	45.8	46.9	37.7	47.0	33.2	35.9	35.5
mammography	84.5	**90.2**	87.3	87.0	42.1	51.7	49.7	39.1
protein_homo	85.6	88.2	**88.9**	85.9	78.7	73.1	51.3	69.8
abalone_19	34.6	59.6	43.6	**63.3**	49.8	42.3	37.7	34.4
Average	61.7	**68.3**	62.8	66.7	60.2	56.8	55.6	56.8

In the same way, **Table 9** provides comparison where split is 6:4. OCFSP shows the best score in 3 datasets and underperforms baseline methods in average AUC score.

Table 9. Comparison where seen data is majority, split is 6:4.

Dataset	Baseline OCC				OCFSP			
	OCSVM	LOF	IF	GMM	DT	LR	GNB	MLP
ecoli	77.0	63.1	77.0	**81.6**	50.2	25.7	39.6	31.9
optical_digits	21.6	**97.3**	45.4	75.6	65.4	51.1	55.5	50.7
satimage	23.1	50.9	34.9	40.3	66.2	48.9	55.5	**69.1**
pen_digits	78.6	**97.7**	81.9	83.9	88.6	77.2	69.7	66.9
abalone	50.0	43.1	55.6	31.8	49.8	69.0	**79.8**	69.1
sick_euthyroid	43.6	**63.0**	40.8	43.7	49.5	39.8	43.1	44.8
spectrometer	82.4	**89.6**	86.3	72.5	69.0	49.6	50.7	55.5
car_eval_34	**97.7**	50.0	89.5	98.3	64.0	91.2	90.4	81.1
isolet	22.3	**76.2**	26.1	37.1	47.6	35.3	46.8	36.7
us_crime	78.9	71.7	**81.1**	75.6	68.9	76.4	57.0	74.0
yeast_ml8	49.4	48.2	49.9	60.2	**55.3**	49.3	50.5	49.5
scene	49.1	**53.4**	47.7	47.3	51.5	43.9	47.1	45.6
libras_move	83.1	88.8	73.6	**95.6**	87.7	90.9	44.3	92.9
thyroid_sick	52.8	**68.1**	48.4	46.1	51.4	60.4	47.9	61.9
coil_2000	54.2	**55.0**	54.6	53.4	50.8	38.9	50.2	37.0
arrhythmia	44.2	42.1	44.3	40.2	46.4	37.3	**49.5**	39.6
solar_flare_m0	75.7	58.8	76.8	**77.4**	62.3	64.9	66.3	65.7
oil	62.7	78.7	76.1	**82.6**	58.7	58.1	49.9	59.9
car_eval_4	**99.1**	50.0	86.3	97.4	59.3	87.3	91.9	77.9
wine_quality	65.4	70.7	70.3	**72.0**	64.2	56.6	62.5	59.0
letter_img	60.2	**91.4**	52.7	52.6	64.7	62.9	53.8	65.5
yeast_me2	**74.9**	63.7	71.2	73.3	49.4	38.2	48.2	33.6
webpage	49.2	71.5	38.3	**89.7**	81.1	77.7	78.1	77.4
ozone_level	**52.8**	45.6	45.6	37.7	45.9	32.4	46.5	35.9
mammography	85.3	**90.4**	87.8	87.0	43.0	49.9	43.1	39.2
protein_homo	86.1	88.6	**88.9**	85.9	77.4	73.2	50.7	68.3
abalone_19	38.4	**67.5**	45.3	63.3	49.8	41.8	40.6	46.1
Average	61.4	**68.0**	62.1	66.7	59.9	56.6	55.9	56.8

Finally, **Table 10** provides the comparison. OCFSP shows the best AUC scores in 6 datasets. As shown the seen data is majority, OCFSP underperforms other OCC methods. Such result is due to the difficulty in feature slide prediction for majority data (see section 5.3).

Table 10. Comparison where seen class is majority, split is 1:9

Dataset	Baseline OCC				OCFSP			
	OCSVM	LOF	IF	GMM	DT	LR	GNB	MLP

ecoli	75.6	77.6	77.9	**81.6**	49.7	26.8	34.7	24.4
optical_digits	23.0	**84.5**	46.5	75.6	63.5	44.1	54.4	56.0
satimage	25.3	57.5	39.5	40.3	**66.5**	47.1	54.4	54.7
pen_digits	77.9	**92.5**	80.4	83.9	85.0	76.3	69.7	75.2
abalone	50.9	42.0	59.3	31.8	49.8	72.3	**80.1**	71.7
sick_euthyroid	43.0	**54.8**	41.4	43.7	48.5	34.2	43.5	44.1
spectrometer	82.0	76.9	**83.5**	72.5	64.8	48.4	55.6	50.6
car_eval_34	78.9	50.3	73.1	**98.3**	59.6	71.0	75.1	72.6
isolet	22.1	**66.4**	24.8	37.1	45.7	35.1	46.2	36.3
us_crime	77.8	76.9	80.6	75.6	76.2	**81.8**	57.9	81.4
yeast_ml8	49.1	52.4	51.0	**60.2**	56.1	50.1	51.1	52.0
scene	49.6	**52.5**	49.8	47.3	**52.5**	41.6	49.2	43.6
libras_move	82.9	51.3	67.1	**95.6**	72.6	72.1	49.4	80.0
thyroid_sick	50.8	52.3	47.7	46.1	49.4	**57.8**	47.9	54.5
coil_2000	**55.0**	50.9	**55.0**	53.4	51.6	39.7	50.1	38.7
arrhythmia	46.3	44.0	42.1	40.2	47.5	39.2	**49.9**	46.0
solar_flare_m0	71.2	71.3	72.4	**77.4**	70.1	66.5	67.5	67.8
oil	59.9	69.2	72.1	**82.6**	66.4	64.5	49.9	65.1
car_eval_4	70.8	50.7	73.7	**97.4**	54.7	68.0	73.2	63.0
wine_quality	66.3	70.5	70.3	**72.0**	65.8	61.7	64.0	57.1
letter_img	60.7	**76.6**	47.4	52.6	58.3	61.7	50.9	49.0
yeast_me2	**78.1**	75.7	75.1	73.3	49.7	39.2	39.5	40.3
webpage	50.1	64.8	39.7	**89.7**	78.6	76.9	81.0	79.4
ozone_level	**54.3**	42.5	44.5	37.7	45.4	30.4	45.6	38.7
mammography	84.8	**88.4**	87.8	87.0	44.3	39.7	44.5	49.8
protein_homo	85.9	**88.3**	88.1	85.9	81.0	72.3	50.5	70.8
abalone_19	42.9	62.9	50.2	**63.3**	49.6	41.6	40.0	39.1
Average	59.8	64.6	60.8	**66.7**	59.4	54.1	54.7	55.6

OCFSP shows comparable performance for minority seen samples. However, the AUC score worsens for majority samples. In particular, AUC scores are less than 50 in several datasets. In which FSP accuracy for unseen minority data is larger than seen majority data. Such problem is related to data distribution of classes. Overall, majority distribution is considered as wider than minority. In such case, learning majority distribution could cover minority distribution. In addition, minority data exists in narrow distribution. In such case, additional data has significant difference from original data. Therefore, FSP task becomes easier.

5 Discussion

This section provides discussion for OCFSP.

5.1 One-class ensemble classifier

In this section, OCFSP is applied as the part of one-class ensemble classifier. In which, OCC classifiers are trained in class by class. Then, all classifiers are combined to create the final classifier. Imbalanced-learn dataset has two classes, minority, and majority. Therefore, minority classifier and majority classifier are trained separately. Then, an ensemble of both OCC classifiers is computed using equations (16) and (17).

$$Score_{ensemble}(X) = Score_{minority}(X) - Score_{Majority}(X) \quad (16)$$

$$f(X) = \begin{cases} minority\ (Score_{ensemble}(X) \geq \lambda) \\ Majority\ (Score_{ensemble}(X) < \lambda) \end{cases} \quad (17)$$

These two equations are based on previous paper [23]. Where λ is threshold value (Deciding such value is not necessary to compute AUC score.). Tables 11-13 compares the ensembled AUC score for OCC and OCFSP.
In particular, **Table 11** provides experiment result where split is 8:2. In which, OCFSP is the best in four datasets. However, such result is not good.

Table 11. Comparison of ensemble classifier, train test split is 8:2.

Dataset	Baseline OCC				OCFSP			
	OCSVM	LOF	IF	GMM	DT	LR	GNB	MLP
ecoli	88.1	95.3	**95.7**	85.5	48.7	89.7	66.7	92.1
optical_digits	33.9	**99.9**	97.8	99.1	98.1	88.7	82.7	90.9
satimage	52.1	85.4	**92.5**	87.0	89.1	73.8	75.0	77.2
pen_digits	86.9	**100.0**	99.4	99.8	99.8	91.2	81.7	98.8
abalone	65.3	75.3	**84.9**	82.8	59.9	71.0	77.2	61.3
sick_euthyroid	45.7	81.2	84.1	**89.0**	65.3	65.7	54.3	79.6
spectrometer	84.2	91.8	**97.0**	49.7	87.9	91.6	80.5	82.5
car_eval_34	**99.8**	98.2	71.2	**99.8**	94.9	98.4	98.8	98.3
isolet	30.2	**99.0**	92.5	93.8	71.0	79.8	62.9	81.2
us_crime	81.5	87.0	**91.3**	74.9	80.9	87.9	78.8	87.4
yeast_ml8	49.5	51.2	54.8	48.0	**59.5**	53.5	52.7	52.9
scene	50.4	67.6	**71.1**	59.0	67.1	65.1	60.0	60.2
libras_move	87.6	92.1	96.6	91.6	94.5	97.7	82.1	**99.2**
thyroid_sick	54.2	64.7	**83.5**	81.7	66.1	79.3	56.1	78.6
coil_2000	54.4	59.9	64.4	66.6	59.6	66.1	53.2	**66.8**
arrhythmia	46.3	41.4	60.5	62.9	67.8	**68.3**	48.6	64.4
solar_flare_m0	77.5	63.3	**80.6**	73.2	62.6	64.9	72.3	60.0
oil	61.6	72.8	76.3	**79.0**	62.0	53.9	50.6	53.8
car_eval_4	**99.9**	97.9	40.6	99.8	98.7	97.6	98.4	98.0
wine_quality	67.2	72.1	**78.4**	76.6	72.0	70.1	67.5	63.7
letter_img	63.1	**99.5**	94.3	98.9	98.6	90.5	70.5	94.7

yeast_me2	79.2	88.3	**89.8**	82.2	51.7	86.9	27.1	88.6
webpage	49.9	**96.7**	49.2	96.9	89.3	91.0	93.6	89.7
ozone_level	57.0	81.3	**85.9**	67.2	69.3	79.6	50.2	79.3
mammography	85.2	**92.4**	91.9	90.4	62.1	86.2	81.0	63.9
protein_homo	85.8	96.8	96.8	**97.4**	83.2	83.3	51.4	84.1
abalone_19	36.5	63.0	72.7	**76.5**	65.6	60.4	44.8	65.7
Average	65.7	**82.0**	81.3	81.8	75.0	79.0	67.4	78.3

Table 12 shows the result where train test split is 6:4. OCFSP is the best in the four datasets.

Table 12. Comparison of ensemble classifier, train test split is 6:4.

Dataset	Baseline OCC				OCFSP			
	OCSVM	LOF	IF	GMM	DT	LR	GNB	MLP
ecoli	85.7	64.5	**95.2**	88.8	52.3	91.2	57.9	92.6
optical_digits	34.1	**99.9**	97.6	99.0	98.1	88.3	82.6	91.1
satimage	51.9	85.1	**93.1**	86.9	88.6	73.5	75.6	77.4
pen_digits	86.9	**99.9**	99.2	99.8	99.8	90.6	81.5	98.8
abalone	63.4	69.6	**84.7**	82.9	62.7	71.9	78.9	64.6
sick_euthyroid	46.5	78.0	80.8	**88.0**	68.0	66.9	54.4	78.3
spectrometer	82.9	89.7	**94.2**	52.8	89.0	91.1	80.7	88.2
car_eval_34	**99.7**	96.6	72.6	99.7	94.8	98.1	98.7	98.2
isolet	29.6	**99.1**	91.4	94.9	72.3	80.0	62.6	80.6
us_crime	80.8	85.9	**91.6**	69.3	80.0	86.6	78.7	86.4
yeast_ml8	50.2	51.3	55.9	47.8	**58.4**	51.2	53.4	51.1
scene	50.3	66.9	**71.8**	58.7	63.3	65.2	58.6	60.8
libras_move	88.5	89.7	94.8	88.4	94.2	**96.0**	81.6	98.8
thyroid_sick	54.7	69.5	**84.9**	82.3	64.7	80.5	56.1	80.1
coil_2000	54.8	58.0	65.5	65.4	58.7	65.9	52.8	**66.6**
arrhythmia	44.8	42.3	51.2	64.7	63.3	**66.8**	49.2	62.1
solar_flare_m0	76.7	59.7	**79.9**	72.1	59.7	68.2	72.8	61.2
oil	63.8	74.5	**80.8**	67.0	64.1	55.5	50.3	53.6
car_eval_4	**99.9**	95.7	49.5	**99.9**	97.7	97.7	98.3	98.1
wine_quality	65.7	73.7	**77.9**	76.5	70.1	69.8	69.4	63.4
letter_img	63.7	**99.2**	93.3	99.0	98.4	90.5	70.8	94.5
yeast_me2	75.7	81.7	**86.2**	84.3	52.4	83.3	30.2	83.9
webpage	49.6	95.1	47.8	**96.6**	88.9	90.9	93.2	89.4
ozone_level	54.7	79.9	**86.6**	73.0	65.8	80.5	50.5	80.1
mammography	85.9	**93.2**	92.5	91.3	66.1	87.6	79.5	65.5
protein_homo	86.3	96.8	97.2	**97.6**	84.2	83.6	51.5	82.6

abalone_19	40.2	69.5	**73.3**	70.1	66.2	62.6	49.1	67.0
Average	65.4	80.2	81.1	**81.4**	74.9	79.0	67.4	78.3

In contrast, **Table 13** provides the result where training testing split is 1:9. In which, OCFSP outperforms other methods in 12 datasets. Moreover, OCFSP-MLP is the best in the average AUC score.

Table 13. Comparison of ensemble classifier, train test split is 1:9.

Dataset	Baseline OCC				OCFSP			
	OCSVM	LOF	IF	GMM	DT	LR	GNB	MLP
ecoli	83.3	89.0	82.4	**90.0**	78.1	83.7	35.1	89.9
optical_digits	35.9	**99.0**	95.9	90.1	92.7	87.8	81.4	88.0
satimage	52.9	90.1	**91.2**	78.6	84.1	74.8	75.1	72.1
pen_digits	86.2	98.6	98.1	**99.7**	99.2	86.5	82.1	97.9
abalone	62.6	72.5	**82.9**	81.7	68.8	76.0	77.8	68.0
sick_euthyroid	46.1	59.6	59.5	84.1	75.4	68.8	52.7	**77.7**
spectrometer	83.1	75.6	83.3	71.1	84.4	74.1	75.4	**88.2**
car_eval_34	95.8	75.5	67.7	**97.4**	83.4	93.8	96.6	92.5
isolet	29.2	**96.2**	87.1	96.1	62.4	80.1	63.7	83.1
us_crime	80.0	79.0	88.1	70.2	81.1	**88.9**	77.2	88.6
yeast_ml8	50.1	53.3	53.6	51.3	**55.9**	51.9	52.8	50.8
scene	50.7	54.0	60.7	58.2	58.9	**64.7**	52.9	58.5
libras_move	**85.4**	45.7	67.1	59.2	77.0	68.0	62.1	80.5
thyroid_sick	53.5	57.6	77.2	78.8	73.5	**81.7**	57.0	80.3
coil_2000	55.6	54.5	65.0	51.1	56.9	64.9	51.6	**66.1**
arrhythmia	46.4	51.6	42.1	56.8	51.3	**59.4**	50.0	55.6
solar_flare_m0	**72.8**	71.1	67.2	61.1	70.4	67.9	71.6	65.9
oil	61.0	70.5	**73.3**	60.0	60.0	54.8	49.2	54.7
car_eval_4	95.5	79.8	61.0	**97.9**	88.8	94.9	96.5	94.5
wine_quality	66.7	70.6	**76.2**	63.3	68.3	67.7	67.9	60.3
letter_img	64.3	96.2	88.5	**97.8**	96.1	92.1	70.5	94.1
yeast_me2	78.8	77.4	82.9	58.2	54.4	**86.5**	39.0	84.7
webpage	50.5	84.8	41.7	82.0	83.5	**88.2**	86.6	88.1
ozone_level	55.9	58.7	74.3	79.5	67.2	**80.3**	47.0	**80.3**
mammography	85.6	90.5	**92.2**	89.7	61.5	74.0	78.2	73.0
protein_homo	86.1	95.0	**96.9**	94.2	87.4	79.2	51.1	77.8
abalone_19	43.6	**62.2**	58.5	54.4	60.9	51.7	54.0	56.1
Average	65.1	74.4	74.6	76.0	73.4	75.6	65.0	**76.6**

Previous three tables show that OCFSP ensemble is relatively effective where training data is small size. Therefore, OCFSP could be considered as one of the alternative OCC algorithms.

5.2 Time Complexity Analysis

Table 14 shows processing time for OCC and OCFSP. In which, webpage dataset is used for time computation. Such dataset has 300 dimensions and is split into 20868 training data (589 minority, and 20279 majority) and 13912 testing data. Comparison is made for both seen and unseen data. In addition, number of the slide Z is 29 for OCFSP. In the OCFSP algorithm, GNB is the fastest FSP model. In contrast, MLP takes time for training.

Table 14. Processing time of OCC algorithms (seconds)

Method	Seen = minority		Seen = majority	
	training	testing	training	testing
OCSVM	0.07	1.0	93	46.9
LOF	0.04	0.3	16.2	9.2
IF	0.3	2.8	5.3	2.7
GMM (1 mixture)	0.05	0.04	1.8	0.04
OCFSP-DT	3.2	2.3	208	2.3
OCFSP-LR	4.7	3.2	204	3.2
OCFSP-GNB	0.4	3.7	31.6	3.7
OCFSP-MLP	42.9	2.3	1624	2.3

Overall, OCFSP takes more time for training stage because FSP model is trained from additional data generated as self-labeled dataset. In which, training time increases where number of the slide Z is large value.

In contrast, OCFSP is fast in the testing stage. Such result is different from previous self-supervised OCC in image dataset. In particular, the previous study reports that classification of geometric transformation task takes time. On the other hand, OCFSP is applicable as real time processing. For this reason, the following hypothesis is considered; 1) Applying feature slide is faster than geometric transformation, 2) Memory size for feature data is smaller than image data. In other words, increasing memory size could provide speed-up for self-supervised OCC in image dataset.

Besides, OCFSP does not change testing time according to the size of training data. Therefore, OCFSP is applicable as real time processing for feature data.

5.3 Accuracy of feature slide prediction subtask

Fig. 4 provides accuracy scores for FSP subtask. Such figure consists of 10 sub-figures. Left sub-figures show the accuracy where seen data is minority. On the other hand, right sub-figures provide the result for majority seen class. In addition, the bottom sub-figures compare accuracy of all applied classification algorithms for seen class. These scores are computed using self-labeled testing set of optical digits dataset.

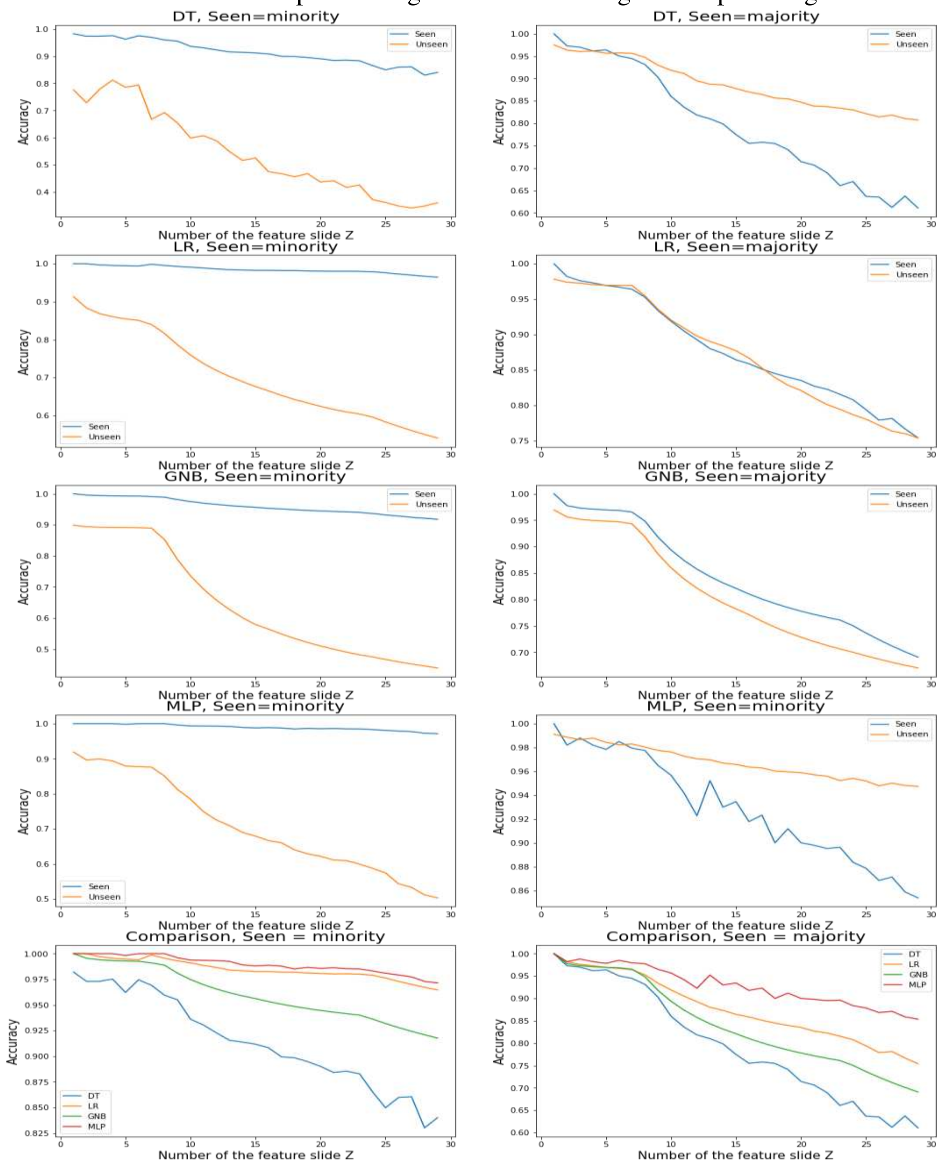

Fig. 4. Feature Slide Prediction accuracy for optical_digits dataset

In which, train test split is done as 8:2. FSP accuracy for seen class is higher than unseen classes, where seen class is minority. In contrast, FSP accuracy does not have difference where seen class is majority. In addition, MLP shows the highest accuracy for seen class.

5.4 Considering other pretext task for feature data.

OCFSP shows fair AUC score for minority data. However, OCFSP is not good where seen data is majority. In which, accuracy for unseen minority data is larger than seen majority data. Such problem is related to data distribution. Overall, majority distribution is considered as wider than minority. Therefore, learning feature slide of majority distribution could cover minority distribution. Such situation is not appropriate for the strategy of the self-supervised OCC in terms of using model error.

In other words, OCFSP is not effective where seen data has diversity. In such case, FSP model learns diverse feature slide and covers FSP task for unseen data. In which FSP model cannot provide model error difference between seen and unseen classes. This is conceptual problem for FSP subtask.

To avoid such problem, other pretext task should be considered for feature data. In which, model should not learn diversity from seen data. The main challenge is how to create such pretext task.

6 Conclusion and Future work

In this paper, OCFSP algorithm is proposed as self-supervised OCC for feature data. The main originality is FSP subtask. In which, the self-labeled dataset is created by sliding feature vectors. Then, this dataset is trained by supervised classification algorithms. Since the training is computed using seen data, accuracy for seen class is considered high relative to unseen class. Accordingly, OCC is computable based on the accuracy of FSP model. Proposed OCFSP is experimented with using an imbalanced-learn dataset and shows comparable performance with other OCC methods. OCFSP provide relatively high accuracy where seen data exists in narrow distribution. Moreover, OCFSP shows consistent testing speed, which is applicable for real time processing. As a weak point, OCFSP takes much time for training. In addition, FSP subtask has limitations in training from wide distribution.

To tackle such weak point, considering other pretext tasks are promising research area. In particular, classification subtask could provide high speed testing time for feature data, in the same way as OCFSP. In which, how to handle diversity of seen data is significant challenge.

Authors Contributions:
The 1st author has done the methodology and implementation coding; the 2nd author is the supervisor and examiner of the whole research project.

Acknowledgements

This study is supported by JSPS KAKENHI (Grants-in-Aid for Scientific Research) #JP20K11955. It is done when the 1st author was working in Iwate Prefectural University. This paper is an extended version of the conference paper published as https://link.springer.com/chapter/10.1007/978-3-030-79463-7_8

The datasets generated during and/or analyzed during the current study are available in [28]

Funding: This work is supported by JSPS KAKENHI (Grants-in-Aid for Scientific Research) #JP20K11955.

Declarations

Conflicts of interest:
The authors declare that they have no conflict of interest.

Ethical approval:
This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A.W.M. van der Laak, Bram van Ginneken, Clara I. Sánchez, A survey on deep learning in medical image analysis, Medical Image Analysis, **2017**, Volume 42, Pages 60-88, https://doi.org/10.1016/j.media.2017.07.005
2. Xufeng Huang, Qiang Lei, Tingli Xie, Yahui Zhang, Zhen Hu, Qi Zhou, Deep Transfer Convolutional Neural Network and Extreme Learning Machine for lung nodule diagnosis on CT images, Knowledge-Based Systems, **2020**, Volume 204, 106230, https://doi.org/10.1016/j.knosys.2020.106230 .
3. Jie Sun, Hui Li, Hamido Fujita, Binbin Fu, Wenguo Ai, Class-imbalanced dynamic financial distress prediction based on Adaboost-SVM ensemble combined with SMOTE and time weighting, Information Fusion, **2020**, Volume 54, Pages 128-144 https://doi.org/10.1016/j.inffus.2019.07.006
4. Fang Liu, Yanwei Yu, Peng Song, Yangyang Fan, Xiangrong Tong, Scalable KDE-based top-n local outlier detection over large-scale data streams, Knowledge-Based Systems, **2020**, Volume 204, 106186, https://doi.org/10.1016/j.knosys.2020.106186.
5. Chandan Gautam, Aruna Tiwari, M. Tanveer,KOC+: Kernel ridge regression based one-class classification using privileged information, Information Sciences, **2019**, Volume 504, Pages 324-333, https://doi.org/10.1016/j.ins.2019.07.052
6. Chandan Gautam, Ramesh Balaji, Sudharsan K., Aruna Tiwari, Kapil Ahuja, Localized Multiple Kernel learning for Anomaly Detection: One-class Classifica-

tion, Knowledge-Based Systems, **2019**, Volume 165, Pages 241-252, https://doi.org/10.1016/j.knosys.2018.11.030

7. Mohammad Saleh Sadooghi, Siamak Esmaeilzadeh Khadem, Improving one class support vector machine novelty detection scheme using nonlinear features, Pattern Recognition, **2018**, Volume 83, Pages 14-33, https://doi.org/10.1016/j.patcog.2018.05.002
8. Mehrnaz Mazini, Babak Shirazi, Iraj Mahdavi, Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms, Journal of King Saud University - Computer and Information Sciences, **2019**, Volume 31, Issue 4, Pages 541-553, https://doi.org/10.1016/j.jksuci.2018.03.011
9. Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Y. Ng. Zero-shot learning through cross-modal transfer. In Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'13). 2013, Curran Associates Inc., Red Hook, NY, USA, 935–943.
10. Scholkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., ¨ and Williamson, R. C. Estimating the Support of a High Dimensional Distribution. Neural computation, **2001**, 13(7): 1443–1471, https://doi.org/10.1162/089976601750264965
11. Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. LOF: identifying density-based local outliers. In ACM sigmod record, **2000**, https://doi.org/10.1145/335191.335388
12. F. T. Liu, K. M. Ting and Z. Zhou, "Isolation Forest," 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 2008, pp. 413-422, https://doi.org/10.1109/ICDM.2008.17
13. Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, Marius Kloft; Deep One-Class Classification. Proceedings of the 35th International Conference on Machine Learning, PMLR 80:4393-4402, 2018.
14. Yang Yang, Chunping Hou, Yue Lang, Guanghui Yue and Yuan He, "One-Class Classification Using Generative Adversarial Networks," in IEEE Access, **2019**, vol. 7, pp. 37970-37979, https://doi.org/10.1109/ACCESS.2019.2905933
15. Izhak Golan, Ran El-Yaniv. 2018. Deep anomaly detection using geometric transformations. In Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18). Curran Associates Inc., Red Hook, NY, USA, 9781–9791.
16. Toshitaka Hayashi, Hamido Fujita, Andres Hernandez-Matamoros, Less complexity one-class classification approach using construction error of convolutional image transformation network, Information Sciences, **2021**, Volume 560, Pages 217-234, https://doi.org/10.1016/j.ins.2021.01.069
17. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. In: ICLR (2018)
18. Long Gao, Lei Zhang, Chang Liu, Shandong Wu, Handling imbalanced medical image data: A deep-learning-based one-class classification approach, Artificial Intelligence in Medicine, **2020**, Volume 108, 101935, https://doi.org/10.1016/j.artmed.2020.101935.
19. Toshitaka Hayashi, Hamido Fujita, One-Class Classification Approach Using Feature-Slide Prediction Subtask for Feature Data, Advances and Trends in Arti-

ficial Intelligence From Theory to Practice 34th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2021 Kuala Lumpur, Malaysia, July 26–29, 2021 Proceedings, Part II, pp. 84-98, https://doi.org/10.1007/978-3-030-79463-7_8
20. Ligang Zhou, Hamido Fujita, Posterior probability based ensemble strategy using optimizing decision directed acyclic graph for multi-class classification, Information Science, **2017**, Volumes 400–401, Pages 142–156 http://dx.doi.org/10.1016/j.ins.2017.02.059
21. Caroline Silva, Thierry Bouwmans, Carl Frélicot, Superpixel-based online wagging one-class ensemble for feature selection in foreground/background separation, Pattern Recognition Letters, **2017,** Volume 100, Pages 144-151, https://doi.org/10.1016/j.patrec.2017.10.034
22. Bartosz Krawczyk, Mikel Galar, Michał Woźniak, Humberto Bustince, Francisco Herrera, Dynamic ensemble selection for multi-class classification with one-class classifiers, Pattern Recognition, **2018**, Volume 83, Pages 34-51, https://doi.org/10.1016/j.patcog.2018.05.015
23. Toshitaka Hayashi., Hamido Fujita. One-class ensemble classifier for data imbalance problems. **2021**, Applied Intelligence, https://doi.org/10.1007/s10489-021-02671-1
24. Paweł Karczmarek, Adam Kiersztyn, Witold Pedrycz, Ebru Al, K-Means-based isolation forest, Knowledge-Based Systems, **2020**, Volume 195, 105659, https://doi.org/10.1016/j.knosys.2020.105659.
25. Toshitaka Hayashi, Hamido Fujita. Cluster-based zero-shot learning for multivariate data. Journal of Ambient Intelligence and Humanized Computing, **2021**, Volume 12, Pages 1897–1911, https://doi.org/10.1007/s12652-020-02268-5
26. Lorenzo Baldacci, Matteo Golfarelli, Davide Lombardi, Franco Sami, Natural gas consumption forecasting for anomaly detection, Expert Systems with Applications, **2016**, Volume 62, Pages 190-201, https://doi.org/10.1016/j.eswa.2016.06.013
27. Ane Blázquez-García, Angel Conde, Usue Mori, Jose A. Lozano, Water leak detection using self-supervised time series classification, Information Sciences, **2021**, Volume 574, Pages 528-541, https://doi.org/10.1016/j.ins.2021.06.015
28. Lemaitre G, Nogueira F, Aridas CK. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. The Journal of Machine Learning Research, **2017**, volume 18, Pages 1-5, https://doi.org/10.5555/3122009.3122026
29. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay; Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, **2011**, 12(85):2825–2830, https://doi.org/10.5555/1953048.2078195
30. Mario A, Figueiredo T, Jain AK, Unsupervised Learning of Finite Mixture Models. IEEE Trans Pattern Anal Machine Intell, **2002**, 24(3):381–396, https://doi.org/10.1109/34.990138