Permuted KPCA and SMOTE to Guide GAN-Based Oversampling for Imbalanced HSI Classification

Tajul Miftahushudur, Bruce Grieve, and Hujun Yin, *Senior Member, IEEE*

***Abstract*—Lack of sufficient and balanced data is one of the major challenges in hyperspectral image classification. This problem can cause poor classification performance, especially in detecting or classifying samples of minority classes. The easiest way to overcome the problem is by resampling or creating synthetic samples to balance the class distributions. As the most advanced generative method, generative adversarial networks (GANs) have been used for generating synthetic data. However, GANs need a large amount or sufficient minority class data to train. In this article, we propose to leverage the synthetic minority oversampling technique (SMOTE) in GANs for creating high quality synthetic data to tackle the imbalance problem. The main idea is to train the generator of the GAN to synthesize data from pattern vectors instead of random noise vectors so to guide the GAN to produce data that can expand the minority class data on the decision boundaries. We used kernel principal component analysis and SMOTE to create the pattern vectors and used a silhouette score to control and prevent overlapping issues. In addition, we applied a self-attention module and an automatic data filter to further minimize potentially wrongly labeled or overlapping samples before being added into the training set. Experimental results on both hyperspectral and remote sensing datasets show that the proposed technique can generate more realistic, diverse, and unambiguous synthetic data, resulting in significantly improved classification performances over the existing oversampling techniques.**

***Index Terms*—Generative adversarial network (GAN), hyperspectral image (HSI), imbalance classification, kernel principal component analysis (kernel PCA), synthetic minority oversampling technique (SMOTE).**

I. Introduction

TREND of implementing multispectral imaging (MSI) and hyperspectral imaging (HSI) in remote sensing applications has been growing recently. These technologies are becoming popular because of their advantages over conventional imaging systems. MSI/HSI can capture up to hundreds of images in various electromagnetic spectral bands, from visible to infrared (400–2500 nm) [1]. These advantages can be utilized to characterize or detect subtle differences or changes in plant physiological traits [2]. As such, they have been applied in various agricultural and food industries, such as plant phenotyping [3], plant health monitoring [4], mapping of yield prediction [5], and quality control of meat or vegetable productions [6].

Information processing on HSI/MSI generally employs machine learning technology due to its accuracy and efficiency in handling vast amount of data. Machine learning needs large and balanced training data to work optimally and to produce high accuracy [7]. Unfortunately, imbalanced data, caused by varying numbers of instances in each class of a dataset, is a natural problem that is difficult to avoid in real-world applications. This problem can affect machine learning model performance, especially when the imbalance ratio is high. Under this condition, samples in the majority class will dominate the model during the training. As a result, the model will tend to classify test samples of a minority/rare class into a majority class. This situation can have severe consequences, such as for health diagnosis, where the machine learning model's ability to detect minority samples (e.g., rare infection cases) is more important than correctly classifying majority samples (healthy cases).

In addition to the challenges that arise during the training phase, it is essential to consider the impact of imbalanced data on the testing phase as well. Imbalanced data in the testing phase can introduce bias in evaluation measures when using standard accuracy metrics. Traditional accuracy metrics, such as overall accuracy (OA) or average accuracy, do not account for varying sample proportions in the testing set. This bias arises because these metrics can yield a high accuracy score even if the model accurately classifies all majority samples but fails to classify any of the minority samples. This high accuracy score does not accurately reflect the model's performance, as it performs poorly on the minority samples. Consequently, previous studies have recommended the use of more appropriate evaluation measures for imbalanced classification problem, such as F-measurement and the Matthew correlation coefficient (MCC) scores [8] [9].

There are various strategies for tackling data imbalance problems to improve machine learning performance. They can be categorized into three levels: data-level, algorithm-level, and hybrid-level [10]. The goal of the data-level approaches is to balance the dataset by incorporating a resampling approach to manipulate and balance the sample numbers of minority and majority classes, so that a conventional classifier can better identify the decision boundary between the majority and minority classes than in the imbalanced conditions [11]. The resampling approach can be further divided into oversampling and undersampling. In oversampling, instances of the minority class are

Manuscript received 12 May 2023; revised 21 July 2023 and 15 September 2023; accepted 3 October 2023. Date of publication 8 November 2023; date of current version 23 November 2023. The work of Tajul Miftahushudur wishes to express gratitude for the scholarship granted by the Indonesian Endowment Fund for Education (LPDP). *(Corresponding author: Hujun Yin.)*

The authors are with the Department of Electrical and Electronic Engineering, The University of Manchester, M13 9PL Manchester, U.K. (e-mail: tajul.miftahushudur@postgrad.manchester.ac.uk; bruce.grieve@manchester.ac.uk; hujun.yin@manchester.ac.uk).

Digital Object Identifier 10.1109/JSTARS.2023.3326963

multiplied until they have equal numbers with the majority class, while undersampling works by removing some instances in the majority class. On the other hand, the algorithm-level methods aim to compensate the domination of the majority class in the training phase by modifying the machine learning algorithm. Algorithm-level can be grouped into three subcategories: threshold moving, cost-sensitive learning, and ensemble learning [12]. The strategy of threshold moving is to change the decision threshold, while the idea of cost-sensitive learning is to add different weights to particular classes [11]. Ensemble learning combines classification outcomes with particular procedures, such as voting and average, from multiple classifier models to obtain one final result. Boosting and bagging [13] are the most popular ensemble techniques. Finally, there is a hybrid-level approach, where a combination of data-level and algorithm-level is used. The purpose of combining two or more methods is to overcome one's weakness with the other's advantages. Therefore, the hybrid technique is more robust than a single data or algorithm level technique. Examples include [14], [15], and [10], which combine data-level and algorithm-level or dual data-level techniques.

Among the three approaches, the data-level is the most popular due to its simplicity. While random oversampling (ROS) and random undersampling (RUS) are the simplest resampling techniques. The idea of ROS is to randomly select samples from the minority class and then duplicate them until the data distribution between the classes is balanced [16]. On the other hand, RUS aims to reduce data of the majority class so to balance the class distribution in the dataset, by randomly selecting and removing data points from the majority class [11]. Although these techniques can effectively decrease the imbalance degree, they have some drawbacks. ROS can be ineffective because the new data that are exactly the same as the original. As the result, it does not add new information to machine learning and can cause overfitting problems. To prevent these, Chawla et al. introduced the synthetic minority oversampling technique (SMOTE) [17], which can minimize the overfitting problem by generating samples that are different from the original training samples.

SMOTE is a widely used and straightforward oversampling technique in addressing imbalanced data. It works by generating new data points between two neighboring instances. This mechanism can introduce diversity to the oversampled dataset and helping minimize overfitting issues. Despite its benefits, SMOTE has a crucial drawback; it does not take into account the presence of the other classes, meaning that SMOTE may generate synthetic samples that overlap with other classes. This overlapping issue can potentially disrupt the decision boundary and lead to less reliable classification results.

Recently generative models become prevalent for data generation. A state-of-the-art technique is the generative adversarial network (GAN). The GAN architecture consists of two networks trained in an adversarial manner, generator, and discriminator. The generator is trained to generate realistic fake data from scratch (from random noise vector input), while the discriminator is a binary classifier trained to distinguish fake data from the generator and real data of the original training set. GAN can generate new data based on the probability distribution of the training data. Therefore, GAN can produce more realistic synthetic data than SMOTE, especially for high-dimensional data [18]. Moreover, GAN can create synthetic data by modifying the attributes, such as shape, color, size, and background, of the original data; hence, it can increase the diversity of the data [19], [20], [21]. Not only for synthesizing images, GANs have also been applied to tasks, such as image super-resolution [22] deblurring [23], denoising [24], and anomaly detection [25].

The ability of GAN in generating realistic synthetic data has shown its potential for solving imbalance classification problems [26]. This has created a new trend in machine learning by replacing the conventional oversampling techniques that are ineffective in synthesising high-dimensional data [18] with a deep learning (DL) model [12], [27]. For the last five years, GAN-based oversampling has been applied in hyperspectral data to overcome imbalance problems. Zhan et al. [28] were one of the pioneers who implemented a GAN for hyperspectral image (HSI) classification. They built a semisupervised classification based on auxiliary conditional GAN (AC-GAN). Then, a further improvement was made by adopting a majority vote concept in [29]. Zhu et al. [30] employed AC-GAN to create synthetic training dataset of HSI. Feng et al. [31] developed a GAN variant consisting of two generators that can generate spectral and spatial data separately and a discriminator to combine the two sets of features. Zhong et al. [32] proposed a method that integrated a conditional random field into the GAN's discriminator. Capsnet-triple GAN [33] integrated Capsnet on triple GAN (that consists of three networks: generator, discriminator, and classifier). Yin et al. [34] integrated dropblock layers on the discriminator. Finally, Roy et al. [35] adopted generative adversarial minority oversampling (GAMO) [36] in spectral or spectral-spatial domain, while existing advanced convolutional neural network (CNN)-based classifiers for HSI data includes 3-D DL [37], 3-D CNN [38], contextual CNN [39], multiscale CNN [40], and Hybrid Spectral-Net (HybridSN) [41], which employ joint spectral-spatial features but do not consider the imbalance conditions.

However, training GANs with existing imbalance data may not be optimal for creating artificial samples of minority classes. This is because that when the available data are imbalance and have a small number of minority-class samples, training GANs may decrease the quality of the synthetic minority data. In addition, without proper fine tuning, there is no guarantee that the generated data would vary enough around the decision boundary.

One of crucial problems during training the GAN is the difficulty in controlling the adversarial training process between the generator and the discriminator. It may increase the potential for gradient descent [42] and causes mode collapse (a condition where GAN repeatedly produces the same data), hence affecting the quality and diversity of the synthetic data. In addition, the ability of the deep networks (generator and discriminator) to learn deep features of complex spectral-spatial data is not always satisfactory, making it difficult for the original GANs to consistently generate good-quality data. Some architectural modifications, such as adding a reconstruction network [43], more discriminators [44] or generators [45], or self-attention

modules (AMs) in the generator and discriminator [46] have been proposed to prevent such problems.

Another main challenge in GAN-based oversampling approach is the difficulty in controlling the distribution of synthetic data within the balanced training set. Generated data have the potential to overlap with the majority class(es). This issue arises because of lack of procedure to consider the presence of majority samples. Involving overlapped data on the balanced training set may lead to problems, such as increased potential for misclassification of majority samples and reduced classifier model performance. Another concern with GAN-based oversampling is that the quality of the generated data that may not realistic and has low diversity. Involving such data into the training set can potentially fail to provide new useful information in enhancing the robustness and generalization of the model.

Recently SMOTified-GAN [47] and DeepSMOTE [48], were developed by combining deep generative model and SMOTE to produce more diversity and plausible instances. SMOTified-GAN employs oversampling based on the SMOTE algorithm as GAN's generator input to help the GAN produce more realistic artificial data than SMOTE. Although on evaluation using imbalance binary UCI datasets, SMOTified-GAN can significantly improve the classification performance, it has limitations. First, it needs to access all the data feature to feed the generator; this mechanism may not be effective for high-dimensional data, such as hyperspectral data as it will drastically burden the generator. Second, the SMOTified-GAN architecture only supports one class oversampling mechanism. As the result, the GAN is not competent to handle multiclass imbalance problems. On the other side, DeepSMOTE is a generative model that employs an autoencoder and SMOTE algorithm to synthesize new data. The encoder and decoder networks in the autoencoder are trained to represent original data in a lower dimensional space and then reconstructed again as close as possible to the original. In order to add diversity, DeepSMOTE trains the autoencoder in a permutation manner and applies SMOTE to the lower dimensional data to increase the quantity of data.

Motivated by the successes of SMOTEfied-GAN and DeepSMOTE in combining SMOTE and GAN while intrigued by the overlapping problems during oversampling in HSI data, we aim to develop a GAN-based oversampling framework that can produce high-quality artificial data by effectively dealing with the decision boundary among imbalance data so that it has the minimum overlap. In the proposed strategy, we employ pattern vectors (PVs) instead of random noise vectors to guide the generator network to train and synthesize new data around the decision boundaries of minority classes. The PVs are obtained by dimension reduction on the original data, followed by oversampling with SMOTE. We used kernel principal component analysis (kernel PCA) to compress the original hyperspectral data that may not be linearly separated. To ensure that PVs are well separated and to prevent overlapping during oversampling with SMOTE, we use silhouette scores to control and determine the best principal components that can separate the PVs well. By training the GAN with a self-AM and utilizing vectors that are well separated from other classes as guide for the generator, we can improve the quality of synthesized data and minimize overlapping. In summary, the proposed framework integrates the advantages of kernel PCA, SMOTE, self-attention, and GAN; hence, it is referred to as synthetic minority PCA-GAN (SMPCA-GAN). Contributions are summarized as follows.

1) We proposed the use of PVs to guide the generator to train and synthesize new data around decision boundaries of minority classes.
2) To enhance the quality of synthetic data, we implemented a self-AM into the generator and the discriminator to capture and learn important spectral or spectral-spatial information of HSI.
3) We developed a filtering protocol based on the silhouette score and inception score to identify poor quality synthetic data and prevent them to be added into the training set.

The rest of this article is organized as follows. Section II describes the background and related work. Section III introduces the proposed method. Experimental results are provided and explained in Section IV. Finally, Section V concludes this article.

II. Background and Related Work

A. PCA and Kernel PCA

PCA is a common technique to reduce data dimensionality. In PCA, coordinates of original dimensional space are linearly transformed onto coordinates of a lower dimensional space. During the conversion, variance of the data plays an important role in deciding the new axes, so to maintain the variance as much as possible.

PCA has advantages of summarizing full featured dataset into a smaller number of variables without eliminating much of the variance or information of the data. Hence, the compressed data can keep the main characteristics of the original dataset. Previous studies have shown that using PCA can improve classification performance [49] and reduce the complexity of machine learning. In general, PCA is done in four steps. The first step is standardization, in which all variables are standardized. As the result, each variable has an equal contribution during analysis. The next step is to compute covariance matrix, followed by calculating eigenvectors and eigenvalues of the covariance matrix to identify the less significant components to remove. Finally, it creates feature vectors with the principal components.

PCA is effective in dimension reduction of linear data. However, hyperspectral data of natural scene is often nonlinear due to nonuniform illumination [50]. Therefore, a nonlinear PCA is necessary. Kernel PCA or KPCA is to map the original data into a higher feature space first via using a kernel function. By using the kernel function in the higher dimensional space, we can obtain the linear principal components more efficiently. As a result, the kernel PCA can effectively separate the data of nonlinear nature in a low-dimensional space.

TABLE I
SMOTE VARIANTS

Borderline SMOTE	Generates new data on the borderline area (area with number of neighborhood minority samples are equal with majority samples)
Safe-level SMOTE	Produce synthetic instance on the safe area (area with number of neighborhood minority samples more than majority samples)
K-means SMOTE	Filtering clusters that only have a few samples before performing oversampling.
SVM SMOTE	Similar to borderline SMOTE, the difference is that the borderline area is determined by SVM, and the minority instances will be generated along the support vector.

When sample x is going to be transformed into the higher feature space with kernel function ϕ, it can be written as $x \rightarrow \phi(x)$. and the mapping process can be solve with dot product operation $K(x_i, x_j) = \phi(x_i)\phi(x_j)^T$. There are several kernel functions that are commonly used to map data into the higher dimensional feature space, for instance:

1) Linear

$$K(x_i, x_j) = x_i^T x_j; \tag{1}$$

2) Polynomial

$$K(x_i, x_j) = \left(\gamma . x_j^T x_j + r\right)^d, \gamma > 0; \tag{2}$$

3) Radial basis function (RBF)

$$K(x_i, x_j) = \exp\left(-\gamma |x_i x_j|^2\right), \gamma > 0; \tag{3}$$

4) Sigmoid

$$K(x_i, x_j) = \tanh\left(\gamma . x_i^T x_j + r\right) \tag{4}$$

where d is degree of polynomial and γ a constant.

B. Synthetic Minority Oversampling Technique

SMOTE is one of the most popular oversampling techniques to handle imbalanced data. It works by randomly selecting a data sample and its k-nearest neighbors; new samples are generated along the lines between the sample and its neighbors. These steps are repeated until the numbers of samples in the minority and majority classes are equal.

One of SMOTE's drawbacks is the overlap problem that causes two different class samples overlap, as illustrated in Fig. 1. As the result, this problem may increase the false-negative rate of a classifier and degrade the performance, especially in detecting the majority samples. The overlap problem occurs because the SMOTE does not consider the presence of majority class samples when generating new minority instances. Therefore, SMOTE can potentially generate synthetic data in the region of majority class. Several modifications of SMOTE have been proposed to tackle this problem, e.g., borderline SMOTE [51], safe-level SMOTE [52], K-means SMOTE [53], and support vector machine (SVM) SMOTE [54]. Table I summarizes the strategies of SMOTE variants.

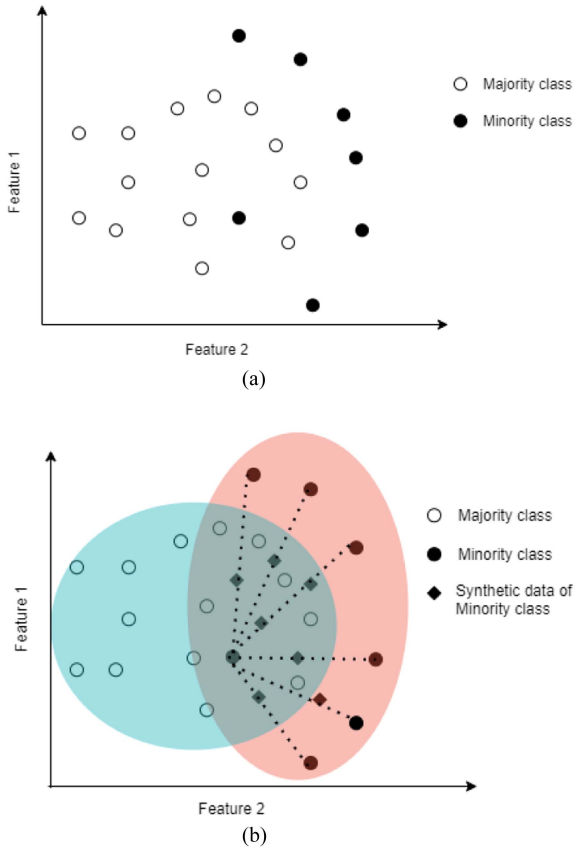

Fig. 1. Illustration of data synthesis by SMOTE: The initial imbalance condition (a), and the possible overlapping problems caused by SMOTE (b).

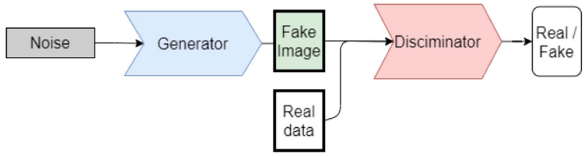

Fig. 2. General architecture of GANs.

C. Generative Adversarial Networks

One of the state-of-the-art generative models is the GAN [55]. Overall, the GAN architecture consists of two networks that competing against each other, generator G and discriminator D. The generator is trained to generate new (fake) instances whilst the discriminator is trained to differentiate fake from real data. The two models are trained together in an adversarial manner. Therefore, the generator can generate data as plausible as possible until the discriminator cannot distinguish the fake and the real.

Fig. 2 illustrates the standard GAN architecture. The generator receives noise input vector z and is trained to transform the noise into a plausible data sample or image $x : G(z) \rightarrow x$, and at the same time, the discriminator, a binary classifier, is trained to distinguish real images (of label 1) and fake ones (of label 0), $D(x) \rightarrow [0, 1]$. Generator loss is calculated based on discriminator feedback and gets a reward if the generator

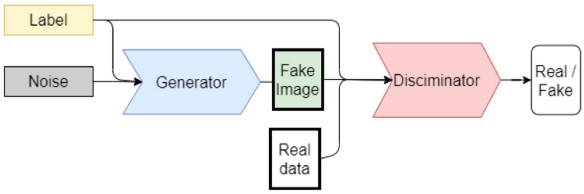

Fig. 3. Architecture of condtional GAN.

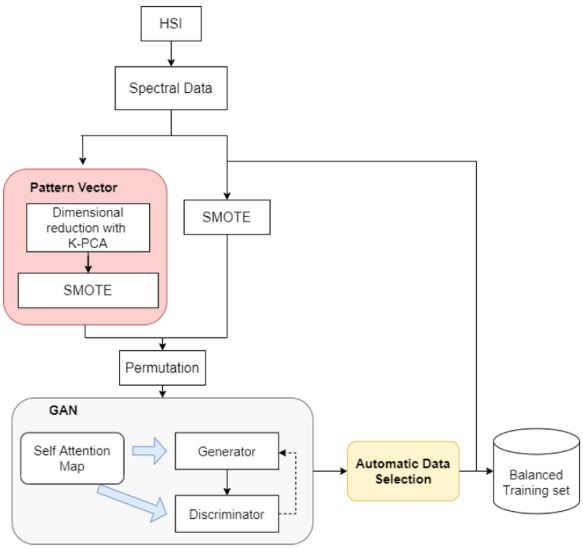

Fig. 4. Illustration of SMPCA-GAN implementation. Self-attention guided GAN is trained to produce realistic samples from PVs and to filter the results before adding them to the training set.

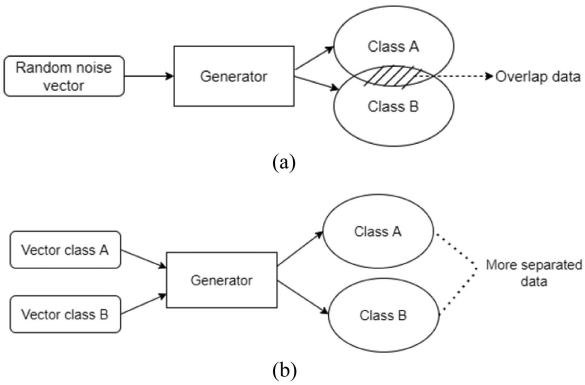

Fig. 5. Data overlapping by regular generator (a), while more separated in proposed generator (b).

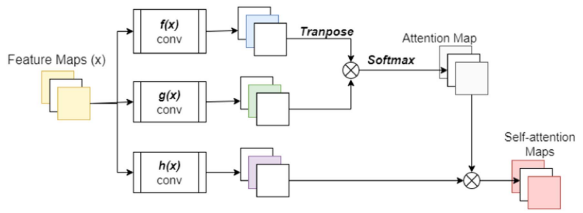

Fig. 6. Self-attention mechanism.

Fig. 7. Examples of UoM dataset shown in RGB image representation.

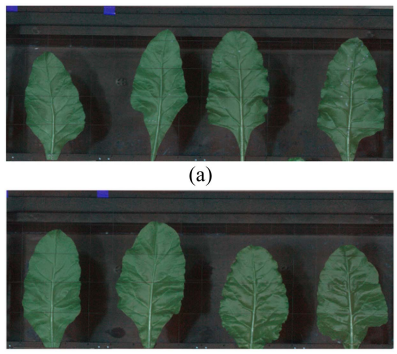

Fig. 8. RGB image representations of Bonn dataset. (a) Cercospora. (b) Rust conditions.

can fool the discriminator. Vice versa, a penalty is incurred if the generator fails to trick the discriminator. On the other hand, discriminator loss will incur a penalty if the discriminator misclassifies a real instance as fake. The objective of GAN is to train the discriminator to maximize (5) and the generator to minimize (6)

$$\frac{1}{m}\sum_{i=1}^{m}[\log(D(x^i)) + \log(1 - D(G(z^i)))] \tag{5}$$

$$\frac{1}{m}\sum_{i=1}^{m}[\log(1 - D(G(z^i)))] \tag{6}$$

where $\log(D(x))$ is the probability discriminator in classifying real data correctly, and $\log(1 - D(G(z)))$ is the probability discriminator classifying synthetic images from the generator as fake correctly. A combination of the two loss functions is called the min–max loss function (7), where the generator tries to minimize this function while the discriminator's goal is to maximize it

$$\min_G \max_D V(D,G) = E_{x\sim pr(x)}[\log(D(x))] + E_{z\sim pr(z)}[\log(1-D(z))]. \tag{7}$$

Although standard GAN can create realistic artificial data, it has a limitation. It is difficult to control a specific object class that one wishes to generate, or in another words, it can only randomly generate data/images. This limitation becomes crucial as in cases of multiclass imbalanced data, where an oversampling model must to be able to create new data for specific classes

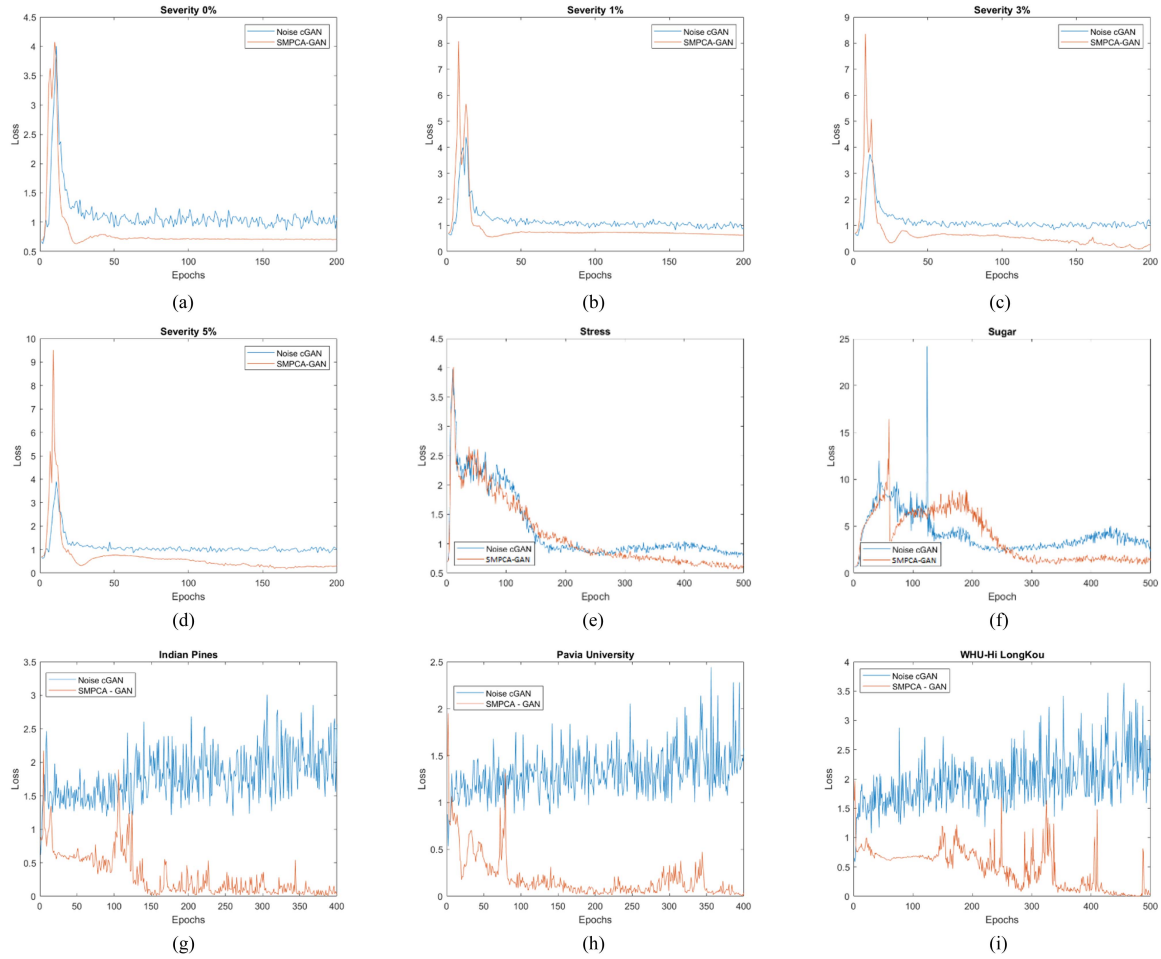

Fig. 9. Generator losses of cGAN and SMPC-GAN on: (a) Bonn-Spec 0% severity, (b) Bonn-Spec 1% severity, (c) Bonn-Spec 3% severity, (d) Bonn-Spec 5% severity, (e) UoM, (f) Bonn, (g) Indian Pines, (h) Pavia Univ., and (i) WHU-Hi LongKou datasets.

such as minority classes. The conditional GAN (cGAN) [56] was developed to overcome the problem. The main idea is to train the generator and discriminator in a conditional manner on a particular class or label. As a result, cGAN can be controlled to generate images based on its input label to the generator. The cGAN architecture is illustrated in Fig. 3.

III. Method

A. Framework of Proposed Method

We take the principle of SMOTE to control the position of generated samples by GAN to be located around the decision boundary. In addition we integrate a self-AM to maintain the consistency of GAN in producing realistic data. Further, we create an automatic data filter module to optimize the quality of the synthetic data with minimal overlapping problems. Flowchart of the proposed SMPCA-GAN is shown in Fig. 4.

Training SMPCA-GAN starts with feature extraction from HSI spectral data. The spectral data are processed further by two separate processes; creating PVs and balancing the data. To create PVs, the spectral data are compressed using kernel PCA and then balanced with SMOTE. The PVs are then fed into the GAN to train the generator. In the other process, the original data is balanced using SMOTE. The data are also fed into the GAN to train the discriminator, specifically to compare the fake samples from the generator based on PV input. During training the GAN, the PVs and the balanced spectral-spatial data are paired by random permutation. Once the training process of GAN is complete and the generator can produce realistic fake data, and the fake samples are evaluated and filtered using an automatic data filter module. This module aims to ensure that the synthesized data are free from overlap problems by filtering out those positioned incorrectly.

The SMPCA-GAN loss function is not much different than the original cGAN. However, there are differences in the generator input and comparing data by the discriminator. Overall, the SMPCA-GAN loss function is described as follows:

$$\min_{G}\max_{D} V(D,G) = E_{x^*}[\log(D(x^*|y))] + E_u[\log(1-D(u|y))] \tag{8}$$

where x^* is the training samples that has been balanced with the use of SMOTE, y is the corresponded label, and u is the compressed data of the training set.

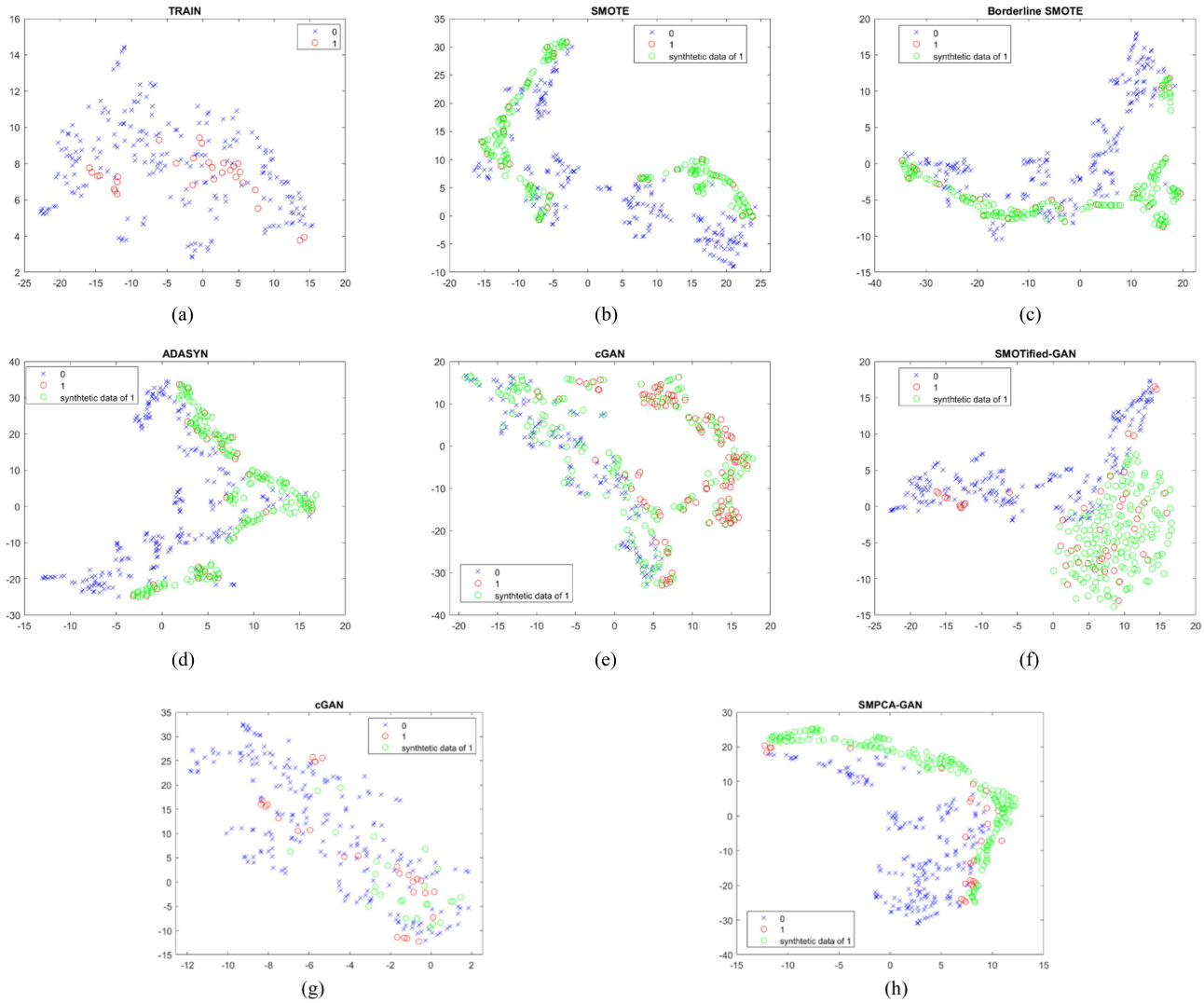

Fig. 10. Visualization of UoM data with t-SNE. (a) Without oversampling. (b) SMOTE. (c) Borderline SMOTE. (d) ADASYN. (e) cGAN. (f) SMOTified-GAN. (g) PCA-GAN. (h) SMPCA-GAN.

Although the proposed SMPCA-GAN appears similar to PCA-GAN [57], with both replacing random noise with compressed training data as the input to the generator, the SMPCA-GAN has three primary differences from the PCA-GAN as follows.

1) Kernel PCA instead of PCA is used to deal with solve nonlinear dimension reduction with HSI data.
2) The use of silhouette score as a control to minimize overlap of the compressed data. Selecting a random number of principle components on kernel PCA can potentially produce overlapping low-dimensional data. As a result, generated data, though realistic, can be mislabeled.
3) The use of SMOTE to generate initial new samples that are completely different from the compressed data of the original samples. This approach is helpful for optimizing the training process when the training data are limited or imbalanced. It can also increase diversity of the synthetic data, compared to PCA-GAN.

B. PVs to Control Synthetic Data

The proposed strategy employs SMOTE to guide GAN in generating realistic and varied data around the decision boundaries of original imbalance data. We employ kernel PCA to represent nonlinear data, such as hyperspectral data in a lower dimensional space. Furthermore, we utilize a silhouette score to determine the number of the best principal components to compress data. Once we obtain the lower dimensional data that are well separated, we implement SMOTE to increase the number of guided data points with less chance of overlapping. We prefer using kernel PCA over autoencoder for dimension reduction for several reasons. First, SMPCA-GAN needs to evaluate how many compressed features that can be effectively oversampled by SMOTE; too many or too few may cause the overlap problem

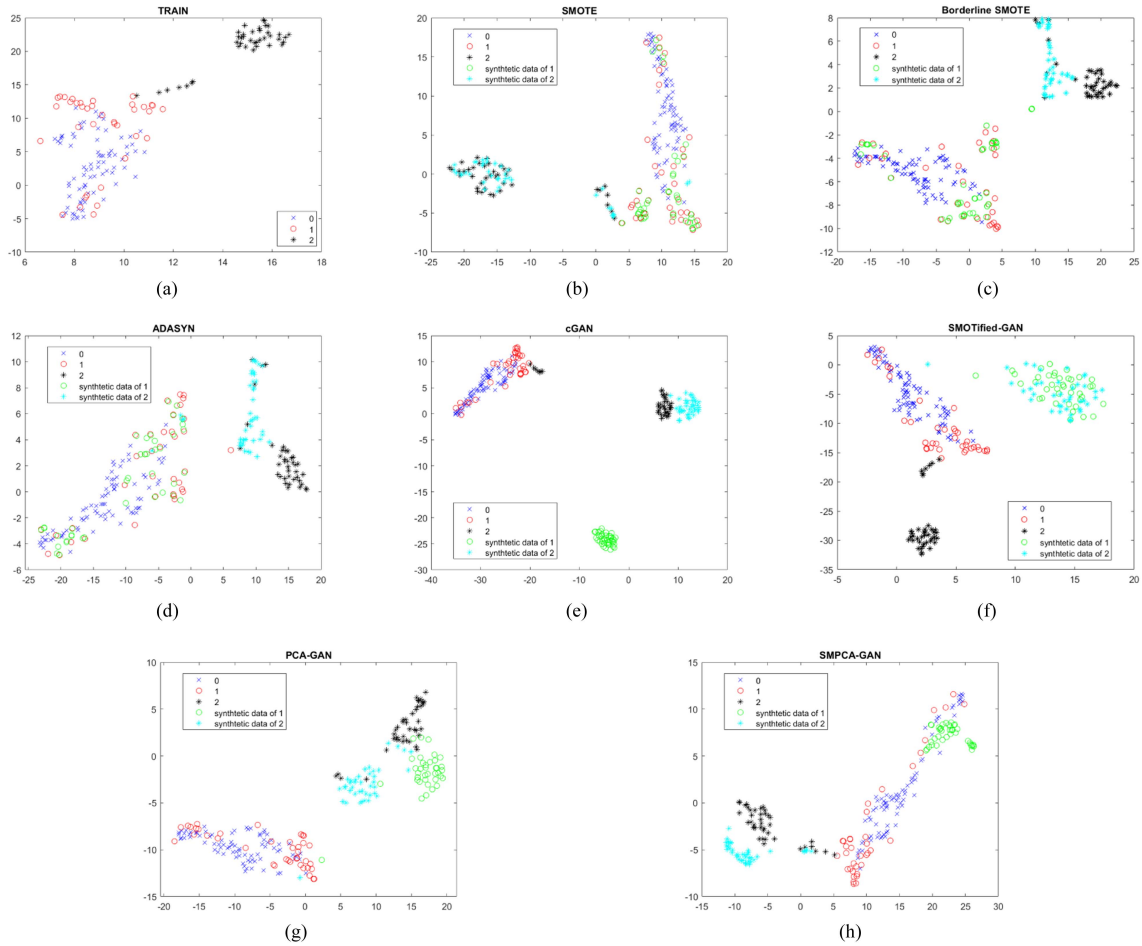

Fig. 11. Visualization of Bonn data with t-SNE. (a) Without oversampling. (b) SMOTE. (c) Borderline SMOTE. (d) ADASYN. (e) cGAN. (f) SMOTified-GAN. (g) PCA-GAN. (h) SMPCA-GAN.

and affect GAN's performance. While autoencoder is more computationally expensive, and its training with many different feature numbers is exhausting and not practical. Second, autoencoder is only effective with a large amount of training data.

We use a silhouette score to find the best number of components that can separate classes well. High silhouette scores indicate that the compressed dataset is well separated. While low scores indicate overlapping, which becomes severer if it is oversampled by SMOTE. Once the compressed data with the best components is obtained, the imbalanced compressed data are balanced with SMOTE before being fed into the generator of the GAN. Intuitively, replacing a random noise vector with a PV can help the generator to produce nonoverlapping samples, as illustrated in Fig. 5.

C. Self-Attention Module

The purpose of implementing a self-AM is to improve the ability of the generator and discriminator to learn important features from the training data. By understanding the significance of the features, the networks can pay more attention on them. The self-AM has been shown to improve the quality of generated images in previous studies [46], [58]. Compared to traditional GANs, integrating the self-AM offers advantages that are not found in traditional GANs. One of the advantages is that the self-attention GAN possesses knowledge of long-range dependencies. For instance, spectral features in green vegetation are known to have higher reflectance in near-infrared bands compared to red bands. Therefore, when generating new data, attention enhanced GAN takes into account such information to generate fake samples and distinguish real and fake samples during its training process.

Fig. 6 shows the mechanism of the self-AM. In general, there are three convolution operations: $f(x)$, $g(x)$, and $h(x)$. These convolution operations extract distinct representation from the input, enabling the module to capture the relationships and dependencies between different parts of the input. The attention maps are computed by measuring the similarity between the output of $g(x)$ and $f(x)$ using dot product operation. The attention maps are then passed through a softmax function to obtain attention weights. The softmax function ensures that the attention weights sum up to 1 and represent the relative importance of different parts of the input. The weights obtained from the softmax are used to compute a weighted sum of the $h(x)$ features. This weighted sum represents the attended features, where more weighting is given to the parts of the input that are deemed important. As the result, the output of this module

TABLE II
TRAINING AND TESTING SAMPLES DISTRIBUTION OF BONN-SPEC DATASET

Disease severity	Healthy (training : testing)	Unhealthy (training : testing)	Total samples
Severity 0%	630 (378: 252)	100 (60 : 40)	730
Severity 1%	630 (378: 252)	100 (60 : 40)	730
Severity 3%	630 (378: 252)	100 (60 : 40)	730
Severity 5%	630 (378: 252)	100 (60 : 40)	730

TABLE III
SUMMARIES OF WHU-HI LONGKOU DATASET

No.	Class	Training (3%)	Testing (97%)	Total
1	Corn	345	34166	34511
2	Cotton	84	8290	8374
3	Sesame	30	3001	3031
4	Broad-leaf soybean	632	62580	63212
5	Narrow-leaf soybean	42	4109	4151
6	Rice	119	11735	11854
7	Water	671	66385	67056
8	Roads and houses	71	7053	7124
9	Mixed weed	52	5177	5229
	Total			204542

TABLE IV
DATASETS

Dataset	Number of instances	Number of bands	Number of classes	Imbalance ratio
Bonn-spec (each severity level)	730	462	2	6.3
UoM	486	33	2	4
Bonn	320	211	3	2
Indian Pines	10249	200	16	122.7
Pavia Univ.	42776	103	9	19.5
Lungkou	204542	270	9	8.3

TABLE V
SUMMARIES OF INDIAN PINES DATASET

No.	Class	Training (30%)	Testing (70%)	Total
1.	Alfafa	14	32	46
2.	Corn-notill	428	1000	1428
3.	Corn-mintill	249	581	830
4.	Corn	71	166	237
5.	Grass-pasture	145	338	483
6.	Grass-trees	219	511	730
7.	Grass-pasture-mowed	8	20	28
8.	Hay-windrowed	143	335	478
9.	Oats	6	14	20
10.	Soybean-notill	292	680	972
11.	Soybean-mintill	736	1719	2455
12.	Soybean-clean	178	415	593
13.	Wheat	62	143	205
14.	Woods	379	886	1265
15.	Build-grass-trees	116	270	386
16.	Stone-steel-towers	28	65	93
	Total			10249

TABLE VI
SUMMARIES OF PAVIA UNIVERSITY DATASET

No.	Class	Training (3%)	Testing (97%)	Total
1.	Asphalt	119	6432	6631
2.	Meadows	599	18050	18649
3.	Gravel	59	2040	2099
4.	Trees	97	2967	3064
5.	Metal sheets	40	1305	1345
6.	Bare soil	154	4875	5029
7.	Bitumen	45	1285	13330
8.	Bricks	104	3578	3682
9.	Shadow	48	889	947
	Total			42776

can captures the important features and relationships within the input data.

In our model, we implement the self-AM into the generator and discriminator networks. Specifically, we place this module between the encoder/feature extraction layers and the output layer in both networks. The implementation of the self-AM in the training process of the generator and discriminator ensures that both networks attend to important features for generating data and classifying fake and real data. Therefore, the generator model can produce fake samples that are closer to the original samples.

D. Data Selection Based on Silhouette and Inception Score

Synthetic data that are useful to improve the classification in imbalance situation should not only be realistic but also have high diversity and low potential to overlap with other classes. Therefore, before the synthetic data are added to the imbalance training set, we evaluate the quality of each sample based on certain criteria.

We use the inception score and silhouette score as quality criteria for the filter to select data. The silhouette score is used to evaluate clustering quality (weather the artificial data is overlapped or mislabeled), while the inception score measures how realistic and diverse the synthetic data are. The data selection process is as follows. Suppose the number of samples needed to balance the original training data is N, then $2\,N$ synthetic samples needs to be generated from the GAN. Each sample is then calculated and ranked based on the combination of the inception and silhouette values. The top N of the synthetics data that have the highest inception and silhouette scores are then added to the training set.

IV. EXPERIMENTS AND RESULTS

A. Performance Measures

It is important to note that in the case of imbalance classification, the use of the accuracy metric may bias the performance measure. Hence, in the experiments, we adopted the F-score and Mathew correlation coefficient (MCC) score to measure the effectiveness of the oversampling methods. While inception score and silhouette score were used to measure the quality of synthetic data.

Fβ-score: Generally Fβ-score can be described as a dynamic mean between precision and recall, where β can be adjusted dynamically. If the recall and precision are equally important, β is set to 1. In case of precision is more important than recall, it can be set to 0.5. Vice versa, β is set to 2 if recall is more important than precision. Equation to calculate Fβ-score is given as

$$F\beta\text{score} = (1+\beta)\frac{\text{Precision}\cdot\text{Recall}}{(\beta^2\text{Precision})+\text{Recall}}. \tag{9}$$

TABLE VII
SILHOUETTE SCORE OF EACH DATASET ON DIFFERENT PCA TYPES

Datasets	Full features	Standard PCA	Kernel type					Hyperparameter	
			Linear	Cosine	Polynomial	RBF	Sigmoid	N component	Gamma
Bonn-spec	0.065	0.069	0.073	0.065	0.073	**0.077**	0.073	4	500
UoM	–0.019	–0.019	–0.01	0.008	–0.011	**0.196**	0.012	6	5000
Bonn	0.213	0.203	0.228	0.219	0.239	**0.279**	0.228	4	1
Indian Pines	–0.009	–0.007	–0.015	–0.013	–0.019	**0.039**	0	26	0.1
Pavia U	0.045	0.046	0.195	0.199	0.161	0.117	**0.216**	6	0.1
Lungkuo	–0.046	–0.032	–0.013	–0.004	–0.021	**0.004**	0	25	10

TABLE VIII
SMPCA-GAN AND CGAN CONFIGURATION

Generator	
Layer	Input shape/value
Input layer	(n kernel PCA components)
Input layer label	(1)
Dense	(128 * n features)
Embedding	(1, n features)
Dense	(1 * n features)
Reshape 1	(n features, 128)
Reshape 2	(n features, 1)
Concatenate	[Reshape 1, Reshape 2]
Conv1D Transpose 1	4 (1 × 512)
Conv1D Transpose 2	4 (1 × 128)
Conv 1D	1 (1 × 1)

Discriminator	
Layer	Input shape/value
Input layer label	(1)
Embedding	(1, 50)
Dense	(1 * n features)
Input layer data	(n features, 1)
Reshape	(n features, 1)
Concatenate	[Input layer data, Reshape]
Conv1D 1	4 (1 × 256)
Conv1D 2	4 (1 × 128)
Flatten	
Dropout	0.4
Dense	1

TABLE IX
HYPERPARAMETER SEARCH

Hyperparameter	Testing
Batch size	32, 64, 128, 256, 512
Learning rate	Between 10^{-2} and 10^{-7}
Activation function	ReLU, leaky ReLU, Tanh

TABLE X
HYPERPARAMETER CONFIGURATION

Dataset	Batch size	Learning rate	Activation function	Epochs	Optimizer
Bonn-spec	64	10^{-5}	Tanh	200	Adam
UoM	64	10^{-6}	Tanh	200	Adam
Bonn	128	10^{-6}	Tanh	200	Adam
Indian Pines	256	10^{-6}	Tanh	400	Adam
Pavia U	128	10^{-5}	Tanh	400	Adam
Lungkuo	128	10^{-5}	Tanh	500	Adam

TABLE XI
3-D SMPCA-GAN CONFIGURATION

Generator	
Layer	Input shape/value
Input layer	(N kernel PCA components)
Input layer label	(1)
Dense	(128 * 7 * 7 * 8)
Embedding	(1, 500)
Dense	(7 * 7 * 8)
Reshape 1	(7, 7, 8, 128)
Reshape 2	(7, 7, 8, 1)
Concatenate	[Reshape 1, Reshape 2]
Conv3D Transpose 1	128 (4 × 4 × 4)
Conv3D Transpose 2	128 (4 × 4 × 4)
Conv3D	1 (7 × 7 × 7)

Discriminator	
Layer	Input shape/value
Input layer label	(1)
Embedding	(1, 150)
Dense	(28 * 28 * 32)
Input layer data	(28, 28, 32, 1)
Reshape	(28,28,32,1)
Concatenate	[Input layer data, Reshape]
Conv3D 1	128 (3 × 3 × 3)
Conv3D 2	128 (3 × 3 × 3)
Flatten	
Dropout	0.4
Dense	1

Matthew correlation coefficient: MCC can be used to summarize a confusion matrix in a single value. In principle, the MCC calculates the correlation between true and predictive scores. Therefore, the higher the correlation, the higher the MCC score. MCC score is formulated as

$$\text{MCC} = \frac{\text{TP}\cdot\text{TN}-\text{FP}\cdot\text{FN}}{\sqrt{(\text{TP+FP})\cdot(\text{TP+FN})\cdot(\text{TN+FP})\cdot(\text{TN+FN})}} \tag{10}$$

where TP is true positive; TN is true negative; FP is false positive; FN is false negative. Inception score: Inception score is a metric to evaluate the quality of generated images or data. This metric can assess how plausible the generated images are and measure their diversity simultaneously. A high inception score indicates high plausible and varied data. The measure is based on the Kullback–Leibler (KL) divergence, obtained by calculating the exponential value of KL divergence, divided by the number of images

$$\text{IS} = \exp\left(\frac{1}{N}\sum_{i=1}^{N} D_{\text{KL}}\left(p\left(y|x^i\right)||\hat{p}(y)\right)\right). \tag{11}$$

Silhouette score: Silhouette score is a metric to measure how well clusters of data are separated. In this experiment, the silhouette score is employed to ensure that synthetic data are well separated from other classes. It has the range between –1 and 1. A high score indicates well separated and clearly distinguishable. When the score is close to 0, it is indicates the distances are

TABLE XII
INCEPTION SCORE

Technique	Bonn-Spec				UoM	Bonn	Indian Pines	Pavia Univ.	WHU-Hi LongKou
	Severity 0%	Severity 1%	Severity 3%	Severity 5%					
SMOTE	1.17 (0.04)	2.03 (0.01)	2.27 (0.02)	5.72 (0.04)	2.78 (0.14)	5.78 (0.15)	6.08 (021)	5.33 (0.14)	1.32 (0.04)
Borderline SMOTE	1.57 (0.02)	1.62 (0.04)	2.29 (0.12)	3.88 (0.04)	2.59 (01.4)	4.91 (0.14)	5.01 (0.18)	8.18 (0.21)	1.53 (0.03)
ADASYN	1.62 (0.04)	1.79 (0.04)	1.42 (0.08)	3.22 (0.10)	4.81 (0.15)	4.81 (0.15)	5.11 (0.19)	8.24 (0.22)	1.64 (0.04)
CGAN	1.02 (0.21)	2.12 (0.8)	3.21 (0.02)	5.56 (0.05)	5.71 (0.14)	5.71 (0.14)	5.41 (0.22)	5.21 (0.17)	1.57 (0.05)
SMOTified-GAN	1.42 (0.12)	2.00 (0.05)	3.64 (0.04)	3.52 (0.06)	3.66 (0.14)	3.66 (0.14)	6.53 (0.20)	8.52 (0.35)	1.33 (0.02)
PCA-GAN	1.22 (0.08)	1.52 (0.07)	2.05 (0.05)	5.21 (0.02)	8.35 (0.14)	8.35 (0.14)	5.25 (0.16)	7.83 (0.16)	1.94 (0.02)
SMPCA-GAN	**1.67 (0.24)**	**2.23 (0.02)**	**4.73 (0.05)**	**6.42 (0.06)**	**9.77 (0.35)**	**9.77 (0.35)**	**8.67 (0.24)**	**8.72 (0.21)**	**2.49 (0.05)**

TABLE XIII
SILHOUETTE SCORE

Technique	Bonn-spec				UoM	Bonn	Indian Pines	Pavia Univ.	WHU-Hi LongKou
	Severity 0%	Severity 1%	Severity 3%	Severity 5%					
SMOTE	0.36 (0.02)	0.28 (0.14)	0.42 (0.04)	0.63 (0.03)	0.06 (0.01)	0.24 (0.01)	0.10 (0.01)	0.20 (0.01)	0.11 (0.12)
Borderline SMOTE	0.31 (0.03)	0.17 (0.02)	0.48 (0.03)	0.42 (0.02)	0.06 (0.02)	0.14 (0.04)	0.07 (0.02)	0.07 (0.03)	0.07 (0.02)
ADASYN	0.32 (0.08)	0.16 (0.04)	0.29 (0.03)	0.42 (0.05)	0.06 (0.02)	0.11 (0.01)	0.09 (0.01)	0.06 (0.01)	0.25 (0.09)
CGAN	0.30 (0.01)	0.28 (0.14)	0.36 (0.02)	0.50 (0.05)	0.10 (0.01)	0.33 (0.04)	0.32 (0.01)	0.18 (0.02)	0.31 (0.12)
SMOTified-GAN	0.53 (0.01)	0.36 (0.10)	0.64 (0.04)	0.65 (0.07)	0.13 (0.01)	0.05 (0.01)	-0.13 (0.01)	-0.01(0.01)	0.09 (0.02)
PCA-GAN	0.42 (0.04)	0.21 (0.12)	0.46 (0.03)	0.40 (0.04)	0.03 (0.02)	0.16 (0.01)	0.11 (0.01)	0.11 (0.01)	0.21 (0.09)
SMPCA-GAN	**0.59 (0.10)**	**0.48 (0.10)**	**0.57 (0.04)**	**0.65 (0.05)**	**0.17 (0.02)**	**0.35 (0.01)**	**0.35 (0.01)**	**0.35 (0.01)**	**0.37 (0.91)**

TABLE XIV
F1 SCORE

Technique	Bonn-spec				UoM	Bonn	Indian Pines	Pavia Univ.	WHU-Hi LongKou
	Severity 0%	Severity 1%	Severity 3%	Severity 5%					
No oversampling	46.32 (0.02)	46.77 (0.07)	46.43 (0.02)	46.33 (0.02)	95.71 (0.05)	85.69 (0.04)	84.51 (0.04)	82.53 (0.04)	94.89 (0.11)
SMOTE	72.59 (0.03)	78.94 (0.02)	93.27 (0.05)	92.75 (0.04)	97.17 (0.05)	84.69 (0.04)	84.39 (0.05)	82.61 (0.04)	95.58 (0.12)
Borderline SMOTE	71.91 (0.06)	76.36 (0.03)	47.42 (0.07)	46.32 (0.05)	97.14 (0.05)	86.53 (0.03)	84.31 (0.06)	82.70 (0.05)	95.56 (0.09)
ADASYN	46.32 (0.02)	73.31 (0.04)	94.04 (0.09)	95.52 (0.04)	97.40 (0.06)	86.37 (0.05)	84.51 (0.03)	82.94 (0.03)	95.51 (0.09)
CGAN	70.42 (0.04)	78.36 (0.12)	86.59 (0.14)	96.41 (0.04)	96.68 (0.05)	85.69 (0.05)	86.40 (0.05)	81.35 (0.03)	94.32 (0.14)
SMOTified-GAN	63.76 (0.12)	57.86 (0.10)	90.13 (0.21)	93.63 (0.05)	96.90 (0.06)	85.69 (0.07)	83.91 (0.05)	82.85 (0.04)	95.92 (0.07)
PCA-GAN	55.15 (0.04)	65.21 (0.09)	63.15 (0.12)	87.01 (0.09)	94.42 (0.06)	85.79 (0.06)	83.42 (0.06)	82.12 (0.8)	96.05 (0.10)
SMPCA-GAN	**72.82 (0.08)**	**84.75 (0.12)**	**95.17 (0.15)**	**98.52 (0.12)**	**98.54 (0.06)**	**87.86 (0.05)**	**87.01 (0.03)**	**83.51 (0.06)**	**96.26 (0.12)**

not significant, while less than 0 shows that there are incorrectly labeled samples. Calculation of the silhouette score is as follows:

$$\mathrm{SS} = (b - a)/\max(a, b) \tag{12}$$

where a is the average distance between samples in the same class (intraclass), and b is the distance between different clusters (interclass).

B. Datasets

In the experiments, we used six hyperspectral datasets: Bonn-spec [59], Bonn University [60], The University of Manchester (UoM) [61], Indian Pines [62], Pavia University [63], WHU-Hi LongKou from Wuhan University [64]. The details of these datasets are presented in Table IV.

The Bonn-spec dataset consists of spectral reflectance of sugar leaves of healthy and unhealthy (cercospora, rust, and powdery mildew). Spectral profiles of controlled and diseased sugar leaves were recorded from day 1 to day 21 with different severity levels (0%, 1%, 3%, and 5%). The dataset has 462 spectral bands that covers spectral range between 400 and 1050 nm with with sampling interval between 1 and 1.4 nm. In this imbalance study, we used a total of 730 samples: 630 healthy and 100 unhealthy. Furthermore, we used 378 instances and 60 instances from the healthy and unhealthy samples or 60% of the samples for training purposes, while the remaining 40% for testing. Table II provides detailed information about training–testing proportion, More comprehensive information regarding the data recording process can be found in [60] and [59].

The UoM dataset contains hyperspectral imagery of arabidopsis of 1024 × 1344 pixels (spatial resolution), covering spectral wavelengths between 400 and 720 nm with 10 nm spectral resolution. There are in total 33 spectral images captured. More detailed information about the hyperspectral instrument can be

TABLE XV
MCC SCORE

Technique	Bonn-spec				UoM	Bonn	Indian Pines	Pavia Univ.	WHU-Hi LongKou
	Severity 0%	Severity 1%	Severity 3%	Severity 5%					
No oversampling	0 (0.0)	0 (0.00)	0.0 (0.0)	0 (0.0)	91.50 (0.02)	77.98 (0.03)	84.08 (0.02)	80.33 (0.02)	90.08 (0.12)
SMOTE	54.45 (0.02)	64.45 (0.04)	87.36 (0.12)	86.50 (0.12)	94.45 (0.02)	75.80 (0.03)	83.89 (0.08)	79.94 (0.02)	92.20 (0.14)
Borderline SMOTE	53.55 (0.03)	52.29 (0.04)	0.0 (0.0)	0 (0.0)	94.38 (0.02)	79.96 (0.05)	83.45 (0.02)	80.18 (0.03)	91.55 (0.05)
ADASYN	0 (0.0)	47.26 (0.02)	88.74 (0.15)	92.29 (0.09)	94.90 (0.02)	78.07 (0.05)	84.08 (0.02)	80.50 (0.02)	90.55 (0.09)
CGAN	49.17 (0.12)	58.02 (0.05)	75.79 (0.21)	95.60 (0.04)	93.41 (0.02)	77.98 (0.03)	86.29 (0.04)	80.25 (0.02)	89.31 (0.07)
SMOTified-GAN	38.63 (0.08)	33.12 (0.05)	81.74 (0.13)	88.05 (0.04)	93.91 (0.02)	77.98 (0.04)	83.83 (0.03)	80.40 (0.05)	92.25 (0.09)
PCA-GAN	21.72 (0.05)	42.21 (0.04)	44.90 (0.21)	76.80 (0.07)	93.63 (0.02)	78.04 (0.04)	83.32 (0.04)	80.32 (0.03)	92.10 (0.13)
SMPCA-GAN	**56.48 (0.09)**	**71.64 (0.08)**	**90.79 (0.20)**	**97.08 (0.09)**	**96.32 (0.03)**	**82.32 (0.04)**	**87.05 (0.05)**	**81.89 (0.07)**	**92.81 (0.15)**

TABLE XVI
CLASSIFICATION PERFORMANCES OF SEVERAL METHODS ON INDIAN PINES DATASET

Class	3-D DL [37]	3-D CNN [38]	3-D contextual CNN [39]	Multiscale 3-D CNN [40]	Hybrid SN [41]	3-D SMPCA-GAN
1	91.81 (0.6)	89.74 (0.46)	93.52 (0.14)	98.42 (0.11)	**100 (0.0)**	**100 (0.0)**
2	92.22 (0.06)	89.71 (0.21)	89.41 (0.12)	98.61 (0.24)	98.07 (0.72)	**98.99 (0.24)**
3	88.89 (0.12)	87.89 (0.25)	89.19 (0.12)	91.82 (0.15)	96.06 (0.46)	**97.32 (0.47)**
4	91.32 (0.08)	92.21 (0.12)	82.52 (0.22)	94.21 (0.17)	**98.36 (1.64)**	97.27 (0.37)
5	96.05 (0.12)	97.42 (0.11)	94.92 (0.12)	96.21 (0.19)	97.18 (1.53)	**98.36 (0.24)**
6	98.84 (0.05)	98.92 (0.13)	97.32 (0.14)	99.04 (0.23)	98.97 (0.74)	**100 (0.0)**
7	96.82 (0.0.2)	97.42 (0.24)	91.91 (0.21)	**100 (0.0)**	**100 (0.0)**	**100 (0.0)**
8	99.54 (0.03)	99.32 (0.21)	99.11 (0.16)	**100 (0.0)**	**100 (0.0)**	**100 (0.0)**
9	83.9 (0.09)	100 (0.22)	73.32 (0.24)	**100 (0.0)**	**100 (0.0)**	**100 (0.0)**
10	91.42 (0.02)	90.71 (0.12)	88.91 (0.21)	97.72 (0.12)	98.36 (0.72)	**99.67 (0.14)**
11	93.43 (0.01)	93.12 (0.11)	92.01 (0.19)	97.52 (0.11)	99.24 (0.08)	**99.62 (0.10)**
12	88.62 (0.25)	95.72 (0.14)	87.52 (0.12)	98.41 (0.19)	99.21 (0.29)	**99.62 (0.27)**
13	99.23 (0.03)	99.73 (0.11)	99.01 (0.11)	**100 (0.0)**	**100 (0.0)**	**100 (0.0)**
14	98.88 (0.05)	99.01 (0.14)	96.92 (0.12)	98.82 (0.12)	99.47 (0.51)	**100 (0.0)**
15	86.82 (0.08)	88.21 (0.09)	81.61 (0.21)	84.92 (0.24)	98.24 (1.53)	**98.50 (0.65)**
16	98.84 (0.08)	98.52 (0.02)	99.21 (0.25)	99.21 (0.08)	**99.26 (1.27)**	85.93 (0.60)
OA	93.01 (0.46)	92.99 (0.15)	91.93 (0.18)	95.97 (0.19)	98.90 (0.17)	**99.3 (0.07)**
F1 score	93.71 (0.43)	93.73 (0.15)	91.87 (0.11)	97.12 (0.11)	98.99 (0.17)	**99.29 (0.07)**
MCC score	92.07 (0.52)	92.05 (0.14)	90.79 (0.11)	95.47 (0.31)	98.81 (0.2)	**99.18 (0.05)**

TABLE XVII
CLASSIFICATION PERFORMANCES OF SEVERAL METHODS ON PAVIA UNIVERSITY DATASET

Class	3-D DL [37]	3-D CNN [38]	3-D contextual CNN [39]	Multiscale 3-D CNN [40]	Hybrid SN [41]	3-D SMPCA-GAN
1	94.32 (0.06)	94.72 (0.04)	94.64 (0.21)	95.55 (0.09)	96.82 (0.2)	**99.97 (0.01)**
2	96.02 (0.03)	96.42 (0.04)	95.02 (0.06)	95.92 (0.08)	99.95 (0.01)	**100 (0.0)**
3	89.23 (0.15)	90.01 (0.07)	81.93 (0.72)	90.24 (0.35)	98.82 (0.02)	**99.47 (0.03)**
4	94.64 (0.05)	95.24 (0.05)	93.73 (0.22)	97.72 (0.05)	92.96 (0.03)	**99.85 (0.09)**
5	99.32 (0.04)	99.02 (0.12)	99.92 (0.02)	99.62 (0.02)	98.31 (0.22)	**99.90 (0.0)**
6	94.34 (0.13)	95.74 (0.23)	83.53 (0.37)	96.42 (0.33)	**100 (0.0)**	**100 (0.0)**
7	88.81 (0.15)	89.21 (0.04)	86.13 (0.21)	92.82 (0.22)	99.54 (0.12)	**100 (0.0)**
8	91.92 (0.10)	92.51 (0.04)	87.73 (0.25)	94.23 (0.12)	95.26 (0.12)	**99.96 (0.01)**
9	98.71 (0.04)	99.33 (0.05)	99.43 (0.06)	99.93 (0.14)	90.64 (0.16)	**100 (0.0)**
OA	93.67 (0.23)	94.23 (0.51)	91.98 (1.65)	94.16 (1.05)	98.25 (0.15)	**99.95 (0.01)**
F1 score	94.69 (0.22)	95.22 (0.51)	91.93 (2.13)	95.72 (1.39)	98.24 (0.23)	**99.94 (0.0)**
MCC score	91.72 (0.33)	92.44 (0.63)	89.35 (2.18)	92.47 (1.07)	97.68 (0.05)	**99.95 (0.01)**

found in [65], [66]. Each image consists six leave samples, where two of them are control, two in heat stress condition, and two in cold stress condition. In total 486 samples were collected and grouped into two classes (control or normal and stressed or abnormal), 54 samples for normal class and 432 for abnormal class. An example of the images in RGB composition is shown in Fig. 7.

The Bonn dataset was captured by a line scanning system that can accommodate 1600 pixels a line. The dataset consists of 200 images over wavelengths ranging from 400 to 1000 nm with 2.8 nm spectral resolution. The detailed information about Bonn system configuration can be found in [59], [60]. Bonn dataset consists of a number of sugar leaves with three different conditions (classes): healthy, infected by Cercospora, and infected by rust, 180 samples of 28×28 spatial patch of the healthy leaves were randomly selected for calculating the average spectral data of each patch. On the other hand, 90 samples of Cercospora and 90 samples of rust were manually selected before being

TABLE XVIII
CLASSIFICATION PERFORMANCES OF SEVERAL METHODS ON WHU-HI LONGKOU DATASET

Class	3-D DL [37]	3-D contextual CNN [39]	Multiscale 3-D CNN [40]	Hybrid SN [41]	3-D SMPCA-GAN
1	84.54 (0.17)	97.51 (0.05)	97.11 (0.03)	98.07 (0.02)	**99.63 (0.02)**
2	63.72 (0.06)	56.34 (0.05)	93.13 (0.09)	99.92 (0.07)	93.10 (0.07)
3	70.32 (0.07)	58.14 (0.15)	51.51 (0.29)	60.83 (0.43)	**83.87 (0.18)**
4	78.42 (0.21)	92.95 (0.04)	95.02 (0.09)	94.13 (0.07)	**99.03 (0.04)**
5	14.01 (0.19)	0.358 (0.49)	83.92 (0.22)	**88.98 (0.09)**	88.12 (0.05)
6	95.92 (0.11)	96.44 (0.16)	98.12 (0.07)	**98.13 (0.03)**	97.47 (0.08)
7	99.72 (0.0)	100 (0.0)	99.52 (0.0)	88.51 (0.16)	**99.92 (0.06)**
8	79.82 (0.03)	78.67 (0.47)	92.81 (0.09)	**89.14 (0.06)**	85.84 (0.03)
9	67.41 (0.91)	39.43 (0.32)	90.42 (0.35)	81.03 (0.04)	**94.07 (0.04)**
OA	85.43 (0.87)	91.69 (0.65)	94.97 (0.67)	95.96 (0.04)	**98.04 (0.07)**
F1 score	85.14 (0.96)	90.46 (0.77)	95.87 (0.84)	95.79 (0.01)	**98.01 (0.08)**
MCC score	82.64 (0.92)	89.09 (0.65)	93.46 (0.83)	95.49 (0.04)	**97.41 (0.10)**

processed as unhealthy samples. An RGB image of the Bonn datasets is shown in Fig. 8.

We also used two remote sensing datasets commonly used in HSI classification studies, the Indian Pines and Pavia University datasets. The Indian Pines contain images of 145 × 145 pixels with a spatial resolution of 20 m/pixel, covering 16 classes of different crops. Originally, the dataset covered 224 spectral bands between 360 and 2500 nm. This number of the band was reduced to 200 after removing bands covering water absorption. On the other hand, Pavia University set has spatial resolution of 610 × 340 pixels and covers 103 spectral bands. Detailed sample distributions of the Indian Pines and Pavia University datasets are described in Tables V and VI, respectively.

Finally, we utilized the WHU-Hi LongKuo dataset, which was acquired in 2018 in the Longkou Town, China. This dataset was collected using an 8-mm focal length Headwall Nano-Hyperspec imaging sensor mounted on a UAV platform flying at an altitude of 500 m. It provides a spatial resolution of 500 × 400 pixels with 270 spectral bands ranging from 400 to 1000 nm. The dataset covers complex agricultural areas, exhibiting nine distinct classes within the scene. Table III provides an overview of the dataset, while detailed information can be found in [64] and [67].

C. Experimental Setting

Creating PVs as the generator input is a crucial step in this framework. It is because that PCA and kernel PCA methods may lead to different performances in data separability. Here, we have tested several kernel PCA methods and results are provided in Table VII.

From the Table VII, it can be seen that kernel PCA with RBF and sigmoid kernels can produce well-separated PVs compared to other kernel types and the PCA, with the highest scores in bold. Therefore, we used these kernel configurations to create PVs.

Next, we revised the cGAN in the proposed SMPCA-GAN architecture. Layer configurations and settings are shown in Table VIII. To obtain an optimal model, fine-tuning the hyperparameters was necessary. Therefore, we list the most important parameters, such as batch size, learning rate, and activation function, that were evaluated, in Table IX. We determined the best parameter configurations using random search procedure by analyzing the quality of generated data based on the inception score and silhouette score; and classification performance based on the MCC score on each possible parameter combination. The best parameter combinations are shown in Table X.

To illustrate the effectiveness of the proposed SMPCA-GAN during the training phase, we plotted the generator loss and compared it with that of cGAN, as shown in Fig. 9. The generator loss of SMPCA-GAN is represented in orange, while cGAN in blue. Overall it can be seen that the SMPCA-GAN is much more effective in reducing the loss, and more stable than cGAN in almost all the datasets, especially for the Bonn-spec dataset. These graphs show that training GAN with structured pattern data as guide can help the generator to converge better than training with unstructured noise. Reliability of the SMPCA-GAN in the training phase also shows its superiority over other GANs, as shown in Fig. 9. Moreover, the SMPCA-GAN appears more stable, with smaller fluctuations as seen in almost all the datasets.

D. Quality of Synthetic Data Results

We compared the proposed SMPCA-GAN with six oversampling techniques: three conventional [SMOTE, borderline SMOTE, and adaptive synthetic (ADASYN)] and three GAN-based (cGAN, PCA-GAN, and SMOTified-GAN). The inception score and silhouette score were used to evaluate the quality of synthesized data.

Tables XII and XIII show the inception scores and silhouette scores, respectively. Overall, both tables reveal that the proposed method yielded better performance against other oversampling techniques. From Table XII it can be seen that SMPCA-GAN produced inception scores higher than other GAN-based or conventional oversampling techniques. On the other hand, Table XIII shows that the SMPCA-GAN produced the highest silhouette scores on all the datasets. Interestingly, SMOTified-GAN produces negative silhouette scores on the Indian Pines dataset and Pavia U datasets, indicating that they produced some mislabeled data.

To help explain how the proposed GAN can outperform other oversampling methods, we visualize the distributions of generated data by each method in 2-D using the t-distributed stochastic neighbor embedding (t-SNE) [68].

Comparisons of data distributions on the UoM and Bonn datasets are shown in Figs. 10 and 11, respectively, with synthesized minority samples marked in green or green and cyan.

As shown in Fig. 10, SMPCA-GAN can produce data with little overlapping between synthesized data of the minority class and samples in the majority class and almost no identical samples in the minority class. As a result, the data are well distributed even in cases where the data appear closely connected to each other. However, synthetic data of SMOTified-GAN appears concentrated within a specific area when part of the original data from the minority samples are excluded. Moreover, there is an overlap between the majority class samples and the synthesized samples. PCA-GAN can only double the original minority data, making it ineffective when oversampling imbalanced data with an imbalance ratio of more than two, such as in the case of the UoM dataset. In the case of cGAN, some oversampled samples are located far from the original minority samples. On the other hand, it appears that neighborhood-based oversampling methods, such as SMOTE, borderline SMOTE, and ADASYN, have some overlapping data and a high degree of similarity.

Similarly for the Bonn dataset, it can be seen that the proposed SMPCA-GAN can produce well-distributed data. The synthetic data of the Cercospora class (cyan) are distributed close to the original data, while the rust class (green) appears to fill the gap in the majority class. In addition, with silhouette score measuring in the high-dimensional space, the generated data show the highest score, hence it can be assumed little overlap. Surprisingly, SMOTified-GAN produced some artificial data that clearly overlapped and are located far from the original data in the 2-D space. As a result, it has the lowest inception and silhouette scores. Oversampling-based cGAN and PCA-GAN also did not perform well, with synthetic rust data located far from the original samples. Therefore, the generated data do not enrich information and expand the original data in classification boundary to potentially improve its classification performance. In addition, there is some overlapping data between Cercospora and rust samples by PCA-GAN. All these indicate that kernel PCA can perform better than the PCA on PCA-GAN in handling nonlinear data. The data distributions of ADASYN and borderline SMOTE look good, whereas the synthetic data of the rust class is well distributed as SMPCA-GAN. However, oversampled data of the Cercospora class looks strange and close to the rust class, making its inception and silhouette scores low, slightly above SMOTified-GAN. In the case of SMOTE, unlike the generated data by SMPCA-GAN that fills the gap of the healthy class (blue), some artificial data of rust class is in touch with the Cercospora samples. From the visualization in Figs. 10 and 11, it can be concluded that the further the distance between synthetic data and original samples, the less realistic the synthetic data. As a result, data may have a low inception score and does not enrich the information and diversity of minority classes to improve classification performance.

The experimental results also show that although SMOTified-GAN and PCA-GAN use feature vectors (full and principal features, respectively) to replace noise as generator inputs, the proposed SMPCA-GAN yields much better performance because that the input vectors have less overlapping issue.

E. Classification Results

Support vector machine (SVM) with RBF kernel was used as the base classifier. Optimal combinations of cost (C) and gamma (γ) parameters were obtained by the grid search method. Each experiment was repeated for thirty times with 5-fold cross validation. We used the F1 score and MCC score to measure classification performance.

Tables XIV and XV show the classification results on the datasets in terms of the F1 score and MCC matrices. Overall, both tables agree that SMPCA-GAN performed significantly better than other oversampling techniques.

To verify whether the proposed method can help tackle multiclass imbalance problems in remote sensing data, the SMPCA-GAN was applied to the Indian Pines, Pavia University, and WHU-Hi LongKou datasets. The experimental results were compared with several previous studies that were developed to handle classification with limited training data. We extended the SMPCA-GAN to synthesize spectral-spatial hyperspectral data in 3-D hypercube format. The 3-D-SMPCA-GAN configuration is described in Table XI.

In the case of spectral-spatial feature extraction and classification, we used CNN with the baseline adopted from HybridSN architecture [41]. The experiment was repeated for five times to obtain satisfactory precision. The preprocessing used the kernel PCA with "RBF" kernel to reduce the dimensions and summarize the spectral features from 200 and 103 to 32, for the Indian Pines and Pavia University datasets, respectively. This step is important to the efficiency of training GANs and reducing memory requirements. Spatial patches of 28×28 pixels from the neighborhood of the centre pixels were extracted as input. As the result, the 3-D voxel input has size of $28 \times 28 \times 32$.

Results on the Indian Pines, shown in Table XVI, indicate that the proposed SMPCA-GAN yielded classification performances better or comparable to the state-of-the-art methods that were specifically designed for limited training data in HSI classification. The performances were measured in several matrices, such as OA, F1 score, and MCC score. Compared to other advance hyperspectral image classification techniques that employ spectral-spatial features, such as [37], [39], [38], [40], and [41], the proposed 3-D-SMPCA GAN gave higher classification performance with improvement around 7% in OA metric and F1 score, and 8% improvement in the MCC score.

Similar results were obtained on the Pavia University and LongKou datasets, as shown in Tables XVII and XVIII. SMPCA-GAN surpassed the baseline classifier with improvements of around 1.7% in OA and F1 score. Compared to other state-of-the-art methods, such as those used in the previous experiments, the proposed method also showed significant differences up to 8% in OA, F1 score, and MCC score. While on Longkou dataset, the proposed oversampling framework surpassed the base classifier by around 2% in OA and MCC score.

TABLE XIX
ABLATION STUDY ON BONN-SPEC, UOM, AND BONN DATASETS

Datasets	F1 score				MCC			
	Base model	+PV	+AM	+AF	Base model	+PV	+AM	+AF
Bonn Spec-Severity 0%	71.67	**74.23**	72.22	73.67	49.72	**56.04**	51.20	49.72
Bonn Spec-Severity 1%	74.9	**85.96**	86.33	86.33	58.40	**75.11**	74.97	74.97
Bonn Spec-Severity 3%	85	**96.17**	90.97	92.78	70.21	**92.66**	81.74	84.93
Bonn Spec-Severity 5%	90.01	**97.75**	90.01	91.88	81.74	**95.60**	83.34	86.50
UoM	95.05	**98.69**	97.43	96.23	93.41	**96.79**	94.87	95.41
Bonn	85.71	**87.42**	86.54	86.32	77.98	**81.21**	79.21	79.01

F. Ablation Study

We conducted an ablation study to evaluate effectiveness of each module, specifically to measure its performance to increase the classification performance. Table XIX summarizes the prediction performances based on additional modules or changes. We used base cGAN as base model, then we added or changed the proposed modules.

Experimental results show that the proposed PV is better than the other modules in improving classification performance, where there is MCC improvement of more than 7% on the Bonn-spec datasets, while on UoM and Bonn datasets it rises up to 3.3%. On UoM dataset, both MCC and F1 scores increase by 1% with the AM and by 2% with the automatic filter (AF), while on Bonn dataset the AM and AF can also increase the scores by 1%. These findings show the usefulness of these modules in improving classification performance.

V. CONCLUSION

In this article, a GAN-based oversampling technique termed SMPCA-GAN is proposed by integrating kernel PCA, SMOTE, and GAN with a self-attention mechanism and an automatic data filter. The proposed strategy is to replace random noise vectors with PVs in the GAN to guide the generator network to synthesize high-quality data. The PVs are obtained by dimensionality reduction through kernel PCA and then oversampling the original data via SMOTE. In this manner, the PVs contain important information of the original data and have a lower dimension, hence avoiding the overlapping problem with SMOTE. Moreover, the GAN is enhanced with a self-AM and filtering procedure to further ensure generated data are free from overlapping.

Compared to the original GAN, which uses random noise as generator input, the SMPCA-GAN trains the generator model more effectively by producing lower loss and better stability. In addition, by using the PVs, it generates new samples around the original ones and expands the decision boundary areas, while preventing potential mislabeled or overlapping samples. The evaluation results indicate that the proposed SMPCA-GAN provided marked improvements over other conventional and GAN-based oversampling techniques, in terms of quality of data and classification accuracy. The evaluation on the benchmark hyperspectral datasets (Indian Pines and Pavia University) also shows that oversampling with SMPCA-GAN can significantly improve classification performance over the state-of-the-art methods that were specifically built for dealing with imbalance hyperspectral data.

Despite the SMPCA-GAN showing much improved performances, several limitations should be taken into account. First, the number of datasets used for evaluation is limited. The use of the self-AM may increase the processing time. Finally, the use of the SMOTE in the proposed method may not be optimal than SMOTE variants, such as borderline-SMOTE, safe-level SMOTE, and ADASYN.

Proposed future work includes integrating SMPCA-GAN with semisupervised learning, so that both generator and discriminator can be used more optimally. Furthermore, updating the combination of SMOTE and kernel PCA with more optimized oversampling and dimensional reduction techniques may also warrant investigation.

ACKNOWLEDGMENT

The authors are thankful to the anonymous reviewers and the editor for their constructive feedback and recommendations, which significantly elevated the manuscript's quality.

REFERENCES

[1] S.-Y. Tan, "Developments in hyperspectral sensing," in *Handbook of Satellite Applications*. New York, NY, USA:Springer New York, 2016, pp. 1–21.
[2] V. Silva-Perez et al., "Hyperspectral reflectance as a tool to measure biochemical and physiological traits in wheat," *J. Exp. Botany*, vol. 69, no. 3, pp. 483–496, 2018.
[3] A. Hennessy, K. Clarke, and M. Lewis, "Hyperspectral classification of plants: A review of waveband selection generalisability," *Remote Sens.*, vol. 12, no. 1, 2020, Art. no. 113.
[4] A. Lowe, N. Harrison, and A. P. French, "Hyperspectral image analysis techniques for the detection and classification of the early onset of plant disease and stress," *Plant Methods*, vol. 13, no. 1, pp. 80–92, 2017.
[5] Y. Z. Feng and D. W. Sun, "Application of hyperspectral imaging in food safety inspection and control: A review," *Crit. Rev. Food. Sci. Nutr*, vol. 52, no. 11, pp. 1039–1058, 2012.
[6] L. Feng et al., "Alfalfa yield prediction using UAV-based hyperspectral imagery and ensemble learning," *Remote Sens.*, vol. 12, no. 12, 2020, Art. no. 2028.
[7] Q. Wei and R. L. Dunbrack, "The role of balanced training and testing datasets for binary classifiers in bioinformatics," *PLoS One*, vol. 8, no. 7, 2013, Art. no. e67863.
[8] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomic.*, vol. 21, no. 1, pp. 1–13, 2020.
[9] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen, "Assessing the accuracy of prediction algorithms for classification: An overview," *Bioinformatics*, vol. 16, no. 5, pp. 412–424, 2000, doi: 10.1093/bioinformatics/16.5.412.
[10] B. W. Yap, K. A. Rani, H. A. Abd Rahman, S. Fong, Z. Khairudin, and N. N. Abdullah, "An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets," in *Proc. 1st Int. Conf. Adv. Data Inf. Eng.*, Singapore, 2014, pp. 13–22.
[11] H. He and Y. Ma, *Imbalanced Learning: Foundations, Algorithms, and Applications*, Hoboken, NJ, USA: Wiley, 2013.

[12] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning From Imbalanced Data Sets*. Berlin, Germany: Springer, 2018.

[13] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, pp. 123–140, 1996.

[14] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost: Improving prediction of the minority class in boosting," in *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*. Berlin, Germany: Springer, 2003, pp. 107–119.

[15] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: A hybrid approach to alleviating class imbalance," *IEEE Trans. Syst., Man, Cybern. Part A: Syst. Hum.*, vol. 40, no. 1, pp. 185–197, Jan. 2010.

[16] P. Branco, L. Torgo, and R. Ribeiro, "A survey of predictive modelling under imbalanced distributions," *ACM Comput. Surveys*, vol. 49, no. 2, pp. 1–50, 2016.

[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321–357, 2002.

[18] R. Blagus and L. Lusa, "Evaluation of smote for high-dimensional class-imbalanced microarray data," in *Proc. - 11th Int. Conf. Mach. Learn. Appl.*, 2012, vol. 2, pp. 89–94.

[19] Y. Peng and H. Yin, "ApprGAN: Appearance-based GAN for facial expression synthesis," *IET Image Process.*, vol. 13, no. 14, pp. 2706–2715, 2019.

[20] A. Bhattacharjee, S. Banerjee, and S. Das, "PosIX-GAN: Generating multiple poses using GAN for pose-invariant face recognition," in *Proc. Lecture Notes Comput. Sci.*, 2019, pp. 427–443.

[21] Q. Li, L. Mai, and A. Nguyen, "Improving sample diversity of a pre-trained, class-conditional GAN by changing its class embeddings," 2019, *arXiv:1910.04760*.

[22] C. Ledig et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 105–114.

[23] J. Su and H. Yin, "Improving adversarial learning with image quality measures for image deblurring," in *Proc. Lecture Notes Comput. Sci.*, 2020, pp. 160–171.

[24] B. Xu and H. Yin, "A slimmer and deeper approach to network structures for image denoising and dehazing," in *Proc. Lecture Notes Comput. Sci.*, 2020, pp. 268–279.

[25] L. Deecke, R. Vandermeulen, L. Ruff, S. Mandt, and M. Kloft, "Image anomaly detection with generative adversarial networks," in *Proc. Lecture Notes Comput. Sci.*, 2019, pp. 3–17.

[26] V. Sampath, I. Maurtua, J. J. Aguilar Martín, and A. Gutierrez, "A survey on generative adversarial networks for imbalance problems in computer vision tasks," *J. Big Data*, vol. 8, pp. 1–59, 2021.

[27] C. Bellinger, R. Corizzo, and N. Japkowicz, "Calibrated resampling for imbalanced and long-tails in deep learning," in *Proc. Lecture Notes Comput. Sci.*, 2021, pp. 242–252.

[28] Y. Zhan, D. Hu, Y. Wang, and X. Yu, "Semisupervised hyperspectral image classification based on generative adversarial networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 212–216, Feb. 2018.

[29] Y. Zhan et al., "Semi-supervised classification of hyperspectral data based on generative adversarial networks and neighborhood majority voting," in *Proc. Int. Geosci. Remote Sens. Symp.*, 2018, pp. 5756–5759.

[30] L. Zhu, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Generative adversarial networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5046–5063, Sep. 2018.

[31] J. Feng, H. Yu, L. Wang, X. Cao, X. Zhang, and L. Jiao, "Classification of hyperspectral images based on multiclass spatial-spectral generative adversarial networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5329–5343, Aug. 2019.

[32] Z. Zhong, J. Li, D. A. Clausi, and A. Wong, "Generative adversarial networks and conditional random fields for hyperspectral image classification," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3318–3329, Jul. 2020.

[33] X. Wang, K. Tan, Q. Du, Y. Chen, and P. Du, "Caps-TripleGAN: Gan-assisted capsnet for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 7232–7245, Sep. 2019.

[34] J. Yin, W. Li, and B. Han, "Hyperspectral image classification based on generative adversarial network with dropblock," in *Proc. - Int. Conf. Image Process.*, 2019, pp. 405–409.

[35] S. K. Roy, J. M. Haut, M. E. Paoletti, S. R. Dubey, and A. Plaza, "Generative adversarial minority oversampling for spectral-spatial hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5500615.

[36] S. S. Mullick, S. Datta, and S. Das, "Generative adversarial minority oversampling," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 1695–1704.

[37] A. Ben Hamida, A. Benoit, P. Lambert, and C. Ben Amar, "3-D deep learning approach for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4420–4434, Aug. 2018.

[38] Y. Li, H. Zhang, and Q. Shen, "Spectral–spatial classification of hyperspectral imagery with 3 D convolutional neural network," *Remote Sens.*, vol. 9, no. 1, 2017, Art. no. 67. [Online]. Available: https://www.mdpi.com/2072-4292/9/1/67

[39] H. Lee and H. Kwon, "Contextual deep CNN based hyperspectral classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2016, pp. 3322–3325.

[40] M. He, B. Li, and H. Chen, "Multi-scale 3D deep convolutional neural network for hyperspectral image classification," in *Proc. IEEE Int. Conf. Image Process.*, 2017, pp. 3904–3908.

[41] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, "HybridSN: Exploring 3-D-2-D CNN feature hierarchy for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 277–281, Feb. 2020.

[42] S. Liu, H. Jiang, Z. Wu, and X. Li, "Data synthesis using deep feature enhanced generative adversarial networks for rolling bearing imbalanced fault diagnosis," *Mech. Syst. Signal Process.*, vol. 163, Art. no. 108139, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327021005197

[43] A. Srivastava, L. Valkov, C. Russell, M. U. Gutmann, and C. Sutton, "VEEGAN: Reducing mode collapse in GANs using implicit variational learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 3308–3318.

[44] T. D. Nguyen, T. Le, H. Vu, and D. Phung, "Dual discriminator generative adversarial nets," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 2667–2677.

[45] Q. Hoang, T. D. Nguyen, T. Le, and D. Phung, "MGAN: Training generative adversarial nets with multiple generators," in *Proc. 6th Int. Conf. Learn. Representations, Conf. Track Proc.*, 2018.

[46] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7354–7363.

[47] A. Sharma, P. K. Singh, and R. Chandra, "SMOTified-GAN for class imbalanced pattern classification problems," *IEEE Access*, vol. 10, pp. 30655–30665, 2022.

[48] D. Damien, B. Krawczyk, and N. V. Chawla, "DeepSMOTE: Fusing deep learning and smote for imbalanced data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 9, pp. 6390–6404, Sep. 2023.

[49] M. Naseriparsa and M. M. R. Kashani, "Combination of PCA with smote resampling to boost the prediction rate in lung cancer dataset," *Int. J. Comput. Appl.*, vol. 77, no. 3, pp. 33–38, 2013.

[50] S. M. Nascimento, K. Amano, and D. H. Foster, "Spatial distributions of local illumination color in natural scenes," *Vis. Res.*, vol. 120, pp. 39–44, 2016.

[51] H. Han, W. Y. Wang, and B. H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced datasets learning," in *Proc. Lecture Notes Comput. Sci.*, 2005, pp. 878–887.

[52] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem," in *Proc. Lecture Notes Comput. Sci.*, 2009, pp. 475–482.

[53] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on k-means and smote," *Inf. Sci.*, vol. 465, pp. 1–20, 2018.

[54] H. M. Nguyen, E. W. Cooper, and K. Kamei, "Borderline over-sampling for imbalanced data classification," *Int. J. Knowl. Eng. Soft Data Paradigms*, vol. 3, no. 1, pp. 4–21, 2011.

[55] I. J. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[56] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784v1*.

[57] C. Wang, P. Wu, J. Fu, Y. Zhou, J. Hu, and L. Yan, "Image classification based on principal component analysis optimized generative adversarial networks," in *Proc. IEEE IDAACS-SWS - 5th Int. Symp. Smart Wireless Syst. Within Int. Conf. Intell. Data Acquisition Adv. Comput. Syst., Proc.*, 2020, pp. 1–4.

[58] W. Zhao, X. Chen, J. Chen, and Y. Qu, "Sample generation with self-attention generative adversarial adaptation network (SaGAAN) for hyperspectral image classification," *Remote Sens.*, vol. 12, no. 5, 2020, Art. no. 843. [Online]. Available: https://www.mdpi.com/2072-4292/12/5/843

[59] A. K. Mahlein, U. Steiner, C. Hillnhütter, H. W. Dehne, and E. C. Oerke, "Hyperspectral imaging for small-scale analysis of symptoms caused by different sugar beet diseases," *Plant Methods*, vol. 8, pp. 1–13, 2012.
[60] A.-K. Mahlein et al., "Development of spectral indices for detecting and identifying plant diseases," *Remote Sens. Environ.*, vol. 128, pp. 21–30, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425712003793
[61] A. Alsuwaidi, B. Grieve, and H. Yin, "Feature-ensemble-based novelty detection for analyzing plant hyperspectral datasets," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 4, pp. 1041–1055, Apr. 2018.
[62] M. F. Baumgardner, L. L. Biehl, and D. A. Landgrebe, "220 band AVIRIS hyperspectral image data set: Jun. 12, 1992 indian pine test site 3," Sep. 2015. [Online]. Available: https://purr.purdue.edu/publications/1947/1
[63] F. Dell'Acqua, P. Gamba, A. Ferrari, J. Palmason, J. Benediktsson, and K. Arnason, "Exploiting spectral and spatial information in hyperspectral urban data with high resolution," *IEEE Geosci. Remote Sens. Lett.*, vol. 1, no. 4, pp. 322–326, Oct. 2004.
[64] Y. Zhong, X. Hu, C. Luo, X. Wang, J. Zhao, and L. Zhang, "WHU-Hi: UAV-borne hyperspectral with high spatial resolution (H^2) benchmark datasets and classifier for precise crop identification based on deep convolutional neural network with CRF," *Remote Sens. Environ.*, vol. 250, 2020, Art. no. 112012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425720303825
[65] D. H. Foster, K. Amano, and S. M. Nascimento, "Color constancy in natural scenes explained by global image statistics," in *Proc. Vis. Neurosci.*, 2006, pp. 341–349.
[66] D. H. Foster, K. Amano, S. M. C. Nascimento, and M. J. Foster, "Frequency of metamerism in natural scenes," *J. Opt. Soc. Amer. A*, vol. 23, no. 10, pp. 2359–2372, 2006.
[67] Y. Zhong et al., "Mini-UAV-borne hyperspectral remote sensing: From observation and processing to applications," *IEEE Geosci. Remote Sens. Mag.*, vol. 6, no. 4, pp. 46–62, Dec. 2018.
[68] L. Van Der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.

Tajul Miftahushudur received the B.Sc. degree in computer science from Diponegoro University, Semarang, Indonesia, in 2013, and the M.Sc. degree in digital image and signal processing, in 2018, from The University of Manchester, Manchester, U.K., where he has been working toward the Ph.D. degree in electrical and electronic engineering, since 2020.

In 2014, he was with the Research Center for Electronics and Telecommunication, Indonesian Institute of Sciences (now known as National Research and Innovation Agency), Bandung, Indonesia, while currently he is temporarily absent from his work for studying in The University of Manchester. His research interests include pattern recognition, machine learning, artificial intelligence, image and signal processing.

Bruce Grieve received the B.Sc. (honors) degree in chemical engineering from the University of York, York, U.K., in 1988, and the Ph.D. degree in process engineering from The University of Manchester, Manchester, U.K., in 2002.

He is a Fellow of the Institute of Engineering and Technology, the Vice President and Fellow of the Institute of Agricultural Engineers, and a Fellow of the Higher Education Academy and holds a Chair with Agri-Sensors and Electronics. He is Co-Founder of the agri-sensors company Fotenix Ltd. Before joining The University of Manchester, he gained 20 years of industrial experience in the fields of online analysis and sensors R&D; including deployment of informatics systems within new integrated products for sustainable agriculture and food. His agri-sensors research focuses on translating those smart technologies in appropriately designed and low-cost forms, to assist with the sustainable delivery of nutritious crops within less developed nations, notably in Sub-Saharan Africa, with the assistance of the Gates Foundation and U.K.-GCRF programmes.

Dr. Grieve has been the Industrial Manager on a number of U.K. Research Council and DTI supported projects. He has held a number of funding board roles with U.K. Research Councils and Innovate-U.K., including being a nominated Member of the BBSRC Agriculture and Food Security Strategy and Policy Panel, and the STFC 21st Century Challenges Strategy Panel.

Hujun Yin (Senior Member, IEEE) received the B.Eng. degree in electronic engineering and the M.Sc. degree in signal processing, in 1983 and 1986, respectively, both from Southeast University, Nanjing, China, and the Ph.D. degree in neural networks from the University of York, York, U.K., in 1997.

Since 1996, he has been with the Department of Electrical and Electronic Engineering, The University of Manchester, Manchester, U.K., where he is currently a Professor of artificial intelligence. He has supervised more than 25 Ph.D. students and authored or coauthored more than 250 peer-reviewed articles. His research interests include neural networks, self-organizing and unsupervised learning, deep learning, image classification and hyperspectral image processing, face recognition, data analytics, time series modelling and prediction, bio-/neuroinformatics, and interdisciplinary applications.

Dr. Yin was a recipient of more than £6 million research funding from U.K. research councils, EPSRC, BBSRC, Innovate U.K., and industries across 25 funded projects. Many of his projects involve industries in developing cutting edge AI solutions. He was an Associate Editor for IEEE TRANSACTIONS ON NEURAL NETWORKS, 2006–2010, and the *International Journal of Neural Systems*, 2005–2020. Since 2015, he has been an Associate Editor for IEEE TRANSACTIONS ON CYBERNETICS. He was also the General Chair or Programme Chair for a number of international conferences in AI, machine learning, and data analytics. Since 2006, he is a Member of the EPSRC Peer Review College, and a Turing Fellow of the Alan Turing Institute, since 2018.