BISSIAM: Bispectrum Siamese Network Based Contrastive Learning for UAV Anomaly Detection

Taotao Li, Zhen Hong, *Member, IEEE,* Qianming Cai, Li Yu, *Member, IEEE,*
Zhenyu Wen, *Member, IEEE* and Renyu Yang, *Member, IEEE*

Abstract—In recent years, a surging number of unmanned aerial vehicles (UAVs) are pervasively utilized in many areas. However, the increasing number of UAVs may cause privacy and security issues such as voyeurism and espionage. It is critical for individuals or organizations to manage their behaviors and proactively prevent the misbehaved invasion of unauthorized UAVs through effective anomaly detection. The UAV anomaly detection framework needs to cope with complex signals in the noisy-prone environments and to function with very limited labeled samples. This paper proposes BISSIAM, a novel framework that is capable of identifying UAV presence, types and operation modes. BISSIAM converts UAVs signals to bispectrum as the input and exploits a siamese network based contrastive learning model to learn the vector encoding. A sampling mechanism is proposed for optimizing the sample size involved in the model training whilst ensuring the model accuracy without compromising the training efficiency. Finally, we present a similarity-based fingerprint matching mechanism for detecting unseen UAVs without the need of retraining the whole model. Experiment results show that our approach outperforms other baselines and can reach 92.85% accuracy of UAV type detection in unsupervised learning scenarios. 91.4% accuracy can be achieved when BISSIAM is used for detecting the UAV type of the out-of-sample UAVs.

Index Terms—UAV anomaly detection, bispectrum, siamese network, unsupervised deep learning, contrastive learning.

1 INTRODUCTION

UNMANNED aerial vehicles (UAVs), aka. drones have proliferated recently and widely adopted in numerous industrial or commercial areas such as weather observation [1], disaster management [2], agricultural irrigation [3], etc. The advancement of such applications is mainly propelled by diverse deep neural networks models [4], [5], [6], [7] and massive-scale high performance computing [8], [9]. While promising, security and privacy issues become the main concerns in the traffic management for the safe presence of UAVs in the airspace [10], [11]. Although the Federal Aviation Administration (FAA) has established laws or policies, defining restricted areas for drone flights, surveillance systems are struggling to keep up and yet ready for the required anomaly detection, particularly when the skies are crowded with a massive spike of UAVs. In such circumstances, *anomalies* are typically referred to as illegal UAV intrusions and anomaly detection aims to effectively and timely identify the UAV types and their flight patterns. To address this, machine/deep learning models [12], [13], [14], [15], [16] are employed to differentiate unknown or illegal drones from the prior whitelist. However, there still exist two interrelated problems:

Complex signal sources in noise-prone environments. Existing work typically collects and extracts UAV signals through analyzing physical signals, such as acoustic [17], [18], radar [19], [20], radio-frequency (RF) signal [21], [22], [23], [24], [25], [26]), or through camera-based target tracking from video streaming [27], [28] or statistically monitoring network traffic data [29], [30]. Acoustic-based approaches are typically sensitive to environmental noises whilst the visual quality of camera is subject to the surrounding conditions such as building blockage, ambient lighting, etc. As opposed to acoustic or vision based techniques, RF signals and traffic data are far less susceptible to environmental factors. Nevertheless, commercial UAVs usually have exclusive communication channel with certain levels of encryption, leading to unprecedented difficulties in acquisition and surveillance. Hence, it is imperative to make the best use of RF signals for anomaly detection in noise-prone environments.

Inadequate samples and limited labeled data. Most approaches assume a huge number of labeled samples acquired and massively used in the supervised model training. However, this assumption can be hardly achieved in a real adversary intrusion scenario, e.g., electronic monitoring or radio hijacking [31] where intruding UAVs operates in a non-cooperative mode within the target area, resulting in the limited access to adequate and labelled samples [32]. This limitation also hinders the quality of unsupervised deep learning models since they are highly dependent upon a large number of samples and particularly ineffective in detection *out-of-sample* entities (i.e., new UAVs beyond the established model and pertaining datasets). Moreover, such models are unsuitable for recognizing and processing RF signal footprints due to its extremely high-dimensional characteristics and time-domain signal dynamics. Given a complex and dynamic environment, it is intricate to precisely capture the potential anomalies or unknown UAVs.

In this paper, we propose BISSIAM, an unsupervised contrastive learning framework based on bispectrum Siamese networks for detecting the UAV intrusions, and the types

- *T. Li, Z. Hong, Q. Cai, L.Yu, Z. Wen are with the School of Cyberspace Security, and the School of Information Engineering, Zhejiang University of Technology, Hangzhou, Zhejiang 310023 China. Email: {2111903074, 2112003088, zhong1983, lyu}@zjut.edu.cn; wenluke427@gmail.com*
- *R.Yang is with the School of Computing, University of Leeds, Leeds LS2 9JT, UK. E-mail: r.yang1@leeds.ac.uk.*

Manuscript received 30 Apr. 2021; revised 1 Aug. 2021; date of current version 1 Aug. 2021 (corresponding authors: Z.Hong and R.Yang)

and operation modes of the UAVs. Firstly, we transform the RF signal into a bispectral amplitude-frequency by leveraging a two-dimensional Fourier transform, which are then transformed into grayscale map and embedded into the Siamese network using an image augmentation strategy. We consider both the symmetry loss of the bispectrum and the cross category mutual information loss $\mathcal{L}_{CMI}$ as the traing objectives to improve the prediction performance of the siamese network and substantially diminish the negative impact of noises on the detection effectiveness. Secondly, we devise a p-sampling based optimization mechanism for selecting the optimal sample proportion involved in model training while ensuring the accuracy of the trained model without compromising the training efficiency. To tackle the out-of-sample detection, we present a similarity-based matching mechanism for pinpointing the proximity between the unseen sample and existing ones, without retraining the whole contrastive model. Experiments show that the bispectral transform in our feature exaction can achieve more than 85% accuracy even in an extremely noise environment and outperform all other feature extractors. The proposed approach can reach 92.85% accuracy of UAV type detection, much higher than other baselines, in unsupervised learning scenarios. When labelled data is available, using BISSIAM to conduct supervised learning can increase the accuracy of detecting the UAV type to 98.57% and the accuracy of detecting the operation mode can also reach 92.31%, which is far higher than other approaches. BISSIAM can also achieve 91.4% accuracy when detecting the UAV type of the out-of-sample UAVs.

This paper makes the following contributions:

- A bispectrum feature extraction from RF signals to be embedded into a siamese network (§3.2).
- An unsupervised contrastive learning framework, with both the symmetry loss of the bispectrum and cross-categories mutual information loss considered, for learning the numerical vector representation (§3.3).
- A novel sampling mechanism to balance the training efficiency and accuracy during the training procedure (§3.4).
- An out-of-sample detection approach that exploits a similarity-based matching algorithm to identify the type of unseen UAVs (§3.5).

Organization. §2 discusses the motivation and main challenges. §3 outlines the overview of BISSIAM followed by the detailed design of the key components. Experiment setup and results are presented in §4 and §5, respectively. §6 discusses the related work before we conclude the paper and future work in §7.

2 MOTIVATION AND CHALLENGES

2.1 Problem Definition

Problem Scope. Fig. 1 showcases a typical use scenario of UAV surveillance system based on radio-frequency signals. The system is deployed to constantly collect and analyze the radio signals so that it can timely perceive any UAVs approaching a target area and detect the types and flight patterns of the intrusive UAVs. In the context of UAV surveillance, any events of blacklisted UAVs and unknown

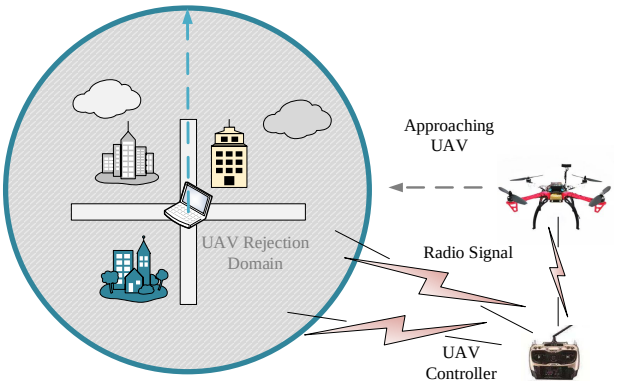


Fig. 1. System scenario.

UAVs manifested in a given management domain are regarded as *anomalies*.

The procedure of *anomaly detection* can be regarded as a series of prediction problem. Formally, we aim to take as input features $\mathcal{X}$ of existing UAVs, to forecast the UAV presence (a binary prediction), the UAV type Y (a K prediction) and the UAV operation mode (aka., flight patterns) Y' of drones (a K' prediction). More specifically, the set of UAV type labels can be defined as $Y = \{y_1, y_2, ..., y_K, y_{non}\}$ while the set of UAV operation mode labels as $Y' = \{y'_1, y'_2, \ldots, y'_{K'}, y'_{non}\}$, where y_{non} or y'_{non} denotes the background noise without any specific UAVs.

Considering the massive RF signal data, the framework should differentiate the required signals in a real wireless environment that contains background noises and other interference from the co-existing radio users such as WiFi and Bluetooth users. Hence, the extracted features $\mathcal{X}$ are obtained on the basis of a collection of RF signals $r(t)$ of the UAV flight and the follow-up signal processing. Without loss of generality, the received RF signal $r_k(t)$ can be defined as:

$$\begin{aligned} r_k(t) &= \phi(e_k(t)) + n(t) \\ &= \alpha \sum_{i=1}^{Re} \beta^e_{k,i} (e_k(t))^i + n(t), k = 1, 2, ..., K+1, \end{aligned} \tag{1}$$

where $e_k(t)$ is referred to as the emitted signal of the k-th UAV controller. α and Re are the wireless channel fading coefficient and the Taylor polynomial order, respectively, while $\beta^e_{k,i}$ denotes the i-th nonlinearity coefficient of the k-th UAV controller. $n(t)$ represents the white Gaussian noise (WGN) with zero mean ($\mu_k = 0$) and variance σ^2_k. Thereafter, signal analysis and processing techniques such as discrete Fourier transform (DFT), short-time Fourier transform (STFT), Hilbert-Huang transform (HHT) are used for obtaining effective features that can be understood by and fed into the prediction models.

Research Challenges. This work addresses three primary research challenges facing the UAV anomaly detection:

Labels are difficult to determine when tackling dynamic signals. The existing supervised learning algorithms rely on a large number of manual labels and particularly a common practice in image processing domain. Doing so on dynamic signals is generally infeasible though – we can hardly determine the corresponding labels directly from observations or signal analysis. For instance, as depicted in Fig. 2, the time-

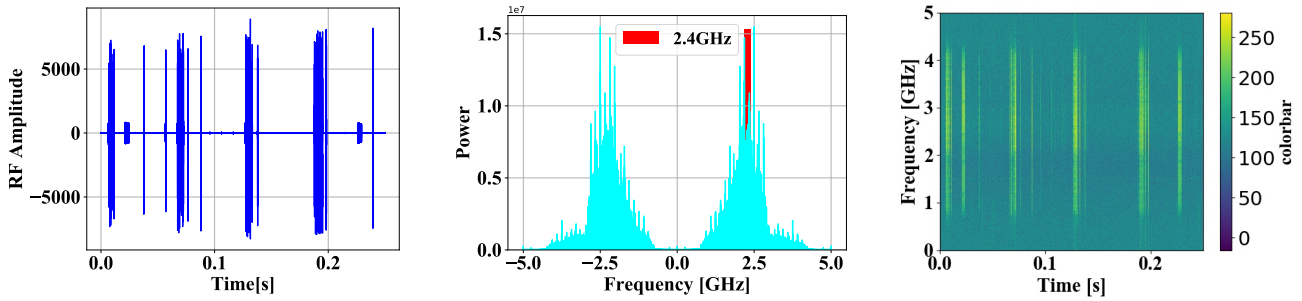


Fig. 2. Parrot AR Drone RF signal: (a) original signal in the *On and Connected* mode; (b) frequency domain diagram after DFT: double peaks at 2.4Ghz, the RF communication band; (c) time-frequency diagram after STFT: huge frequency variations over time.

domain diagram of the original Parrot AR Drone RF signal (Fig. 2a) is converted into a frequency domain diagram through DFT and STFT (Fig. 2b and 2c). However, no UAV label information can be intuitively acquired, reserved and calibrated in the process of DFT and STFT. The non-cooperative mode in most intrusive UAVs also makes it difficult to obtain sufficient high-quality labeled data and to allow further the examination of the flight logs for any data calibration. Therefore, it calls for an effective unsupervised learning mechanism for coping with the characteristics of RF signals.

A balance to strike between the sample size and the accuracy of unsupervised learning. Although training upon massive samples could intrinsically result in higher model precision, either sample collection or model construction is time/resource-consuming and susceptible to model updates that are norm rather than the exception in the ever-growing presence of unseen UAVs. Hence, it is highly desirable to learn from a moderate number of UAV samples to improve the efficiency of model training and maintenance.

Out-of-sample anomaly detection. The surveillance system is increasingly exposed to a variety of threats and attacks from unknown UAVs, e.g., to photograph a new piece of environment. General machine learning classification systems perform the detection task by setting a pre-determined number of categories, e.g. using $softmax$ for category probability calculation. The effectiveness of detecting unknown drones is substantially susceptible to the misclassification. Anomaly identification should allow forecasting new UAVs and rapidly catching the up-to-date malicious intrusions.

2.2 Unsupervised Deep Learning

Traditional unsupervised learning techniques such as K-means [12] cannot be easily applied in high-dimensional data due to a catastrophe of dimension [33]. Recent work on unsupervised deep learning is divided into the following notable aspects: generative learning, e.g., generating adversarial networks (GAN) [34], variational auto-encoder (VAE) [35]) and contrastive learning such as simple framework for contrastive learning of visual representations (SimCLR) [36] and simple siamese (SimSiam) [37].

Generative models, mainly based on GAN and auto-encoder (AE) architectures, aim to generate information with high-level semantics from data to assist the unsupervised categorization. For instance, information generating adversarial networks (info-GAN) [14] can generate attribute

TABLE 1
Important Symbol Notations

Notation	Description
$r_k(t)$	the k-th UAV RF signal
Y	UAV label
R^k_{3r}	the third order cumulant of the signal $r_k(t)$
B^k_r	bispectrum of the signal $r_k(t)$
$\mathcal{X}$	grayscale image of bispectrum
S_p	a sample subset with the proportion p
$\mathcal{T}$	image transformation set
Z	the collection of mapping vectors
V	the collection of encoding vectors
$I(x, y)$	the mutual information of x and y
J	cost function
W	adjacency matrix to store the node similarity

information (like category, shape, size, etc.) from images. However, GAN networks are prone to pattern collapse and VAE heavily relies on the pre-training level of auto-encoder. Generative models focus on complicated information details of the image, and hence the optimization is of even-increasing difficulty. By contrast, contrastive learning leverages the image comparison to bring features of analogous classes much closer to each other so that similar samples stay close to each other while dissimilar ones are far apart. In a formalized manner, contrastive learning learns a mapping function f and encodes the data x into a feature space $f(x)$ such that

$$Sim.(f(x), f(x^+)) >> Sim.(f(x), f(x^-)), \tag{2}$$

where x^+ denotes a sample similar to x and x^- denotes a sample not similar to x, while $Sim.$ represents the similarity score. In effect, one only needs to ensure that the similarity $Sim.(f(x), f(x^+))$ within the same class is much larger than $Sim.(f(x), f(x^-))$ of different classes. Contrastive learning can observably outperform other supervised learning approaches on some datasets [36], [37], [38], [39].

This work will focuses on developing a rapid and stable unsupervised framework based on contrastive learning, which is simpler and faster to optimize with more stable accuracy as opposed to generative models.

3 Proposed Approach of BisSiam

3.1 Overview

We present a siamese network based contrastive learning framework for unsupervised representation learning and the subsequent anomaly detection. Fig. 3 depicts the detailed pipelines of the framework.

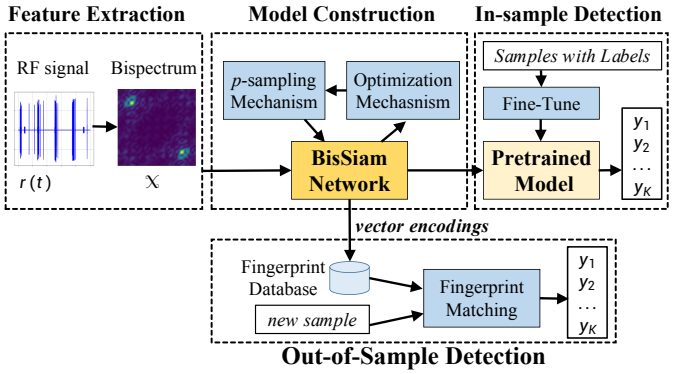


Fig. 3. The workflow of BISSIAM

Initially, the feature extraction module transforms the original UAV RF signal into a bispectrum, via Fourier transform, which is then fed into the model construction (§3.2). We then devise the BISSIAM network, the cornerstone of the contrastive learning framework, to generate the encoded representation of UAV samples and learn the UAV types and the flight patterns. By using the encoded representations and few samples of drones with labels, the pre-trained model will be fine-tuned and then used for the in-sample anomaly detection (§3.3). Meanwhile, we propose a sampling mechanism for minimizing the sample size involved in model training while ensuring the accuracy of the trained model. This can accelerate the model training without compromising the model accuracy (§3.4). To tackle the unseen UAVs, we present a similarity-based fingerprint matching mechanism for pinpointing the proximity between the unseen one and the encoded vectors of the existing samples, without the need for retraining the whole contrastive learning model, which is time-consuming (§3.5). Table 1 demonstrates the symbolic definitions of variables used in the following context.

3.2 Feature Extraction

Fourier transform. High-dimension and dynamicity of the time domain signals complicate the procedure of learning representation directly from raw signal data. To obtain the spectrum, signal processing, e.g., fast Fourier transform (FFT), is typically leveraged to extract the features from original signals. Consequently, the spectrum of similar signals will have similar features and can be easily recognized and processed by a convolutional neural networks (CNN).

To make it clear, as shown in Fig 2a, we need to transform over 10 million sampling points of the original time domain signal $r_k(t)$ into the frequency domain signal $r_k(f)$ by one-dimensional Fourier transform as follows:

$$r_k(f) = \int_{-\infty}^{+\infty} r_k(t) e^{-j2\pi ft} dt, \tag{3}$$

where f, j and dt denote the frequency, the imaginary symbol and the differential, respectively. However, one-dimensional Fourier transformation usually comes with loss of information and only retains the the information in the frequency domain. The identification of RF signal is also susceptible to White Gaussian Noise (WGN) widely manifesting in the air [21] and one-dimensional Fourier transform cannot eliminate the WGN, and hence has intrinsic limitations.

Bispectrum feature extraction. To extract more frequency domain features, we adopt a bispectrum $B_r^k(f_1, f_2)$ based on two-dimensional Fourier transform:

$$B_r^k(f_1, f_2) = \int_{m_1} \int_{m_2} R_{3r}^k(m_1, m_2) e^{-j2\pi(f_1 m_1 + f_2 m_2)} dm_1 dm_2, \tag{4}$$

where f_1 and f_2 denote the frequency bins which represent the frequency domain information corresponding to the delay m_1 and m_2 in the time domains. $R_{3x}^k(m_1, m_2)$ indicates the third order cumulant or the skewness, i.e., the degree of symmetry between the signal distribution and the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ of the signal $r_k(t)$:

$$R_{3r}^k(m_1, m_2) = \mathbb{E}[\overline{r_k(t)} r_k(t + m_1) r_k(t + m_2)], \tag{5}$$

where $\mathbb{E}[\bullet]$ denotes the math expectation and $\overline{\bullet}$ denotes the conjugate operation.

In fact, the bispectrum provides more comprehensive information as opposed to the general Fourier transform, including the two-dimensional frequency domain (f_1, f_2) of the crossover, and the skewness statistical properties between the crossover frequencies. In particular, the bispectrum can effectively mitigate the impression of WGN because the WGN has a mean value of 0, which can be filtered by the expectation operation $\mathbb{E}[\bullet]$ in Eq. 5. Another reason for using the bispectrum is because the two frequency domain dimensions (f_1, f_2) and the frequency domain amplitude $|B(f_1, f_2)|$ can be directly transformed into grayscale maps and fed into a CNN. The frequency domain amplitude characteristics $|B(f_1, f_2)|$ can be calculated as follows:

Lemma 1. If the k-th signal is a finite length sequence $h_k(t)$, its bispectrum [32] is

$$B_h^k(f_1, f_2) = H_k(f_1) H_k(f_2) \overline{H_k}(f_1 + f_2), \tag{6}$$

where $H_k(f) = \sum_t h_k(t) e^{-j2\pi ft}$. By Lemma *1* we can obtain

$$\begin{aligned} B_h^k(f_1, f_2) &= \left|B_h^k(f_1, f_2)\right| e^{j\varphi_k(f_1, f_2)}, \\ H_k(f) &= |H_k(f)| e^{j\varphi_k(f)}, \end{aligned} \tag{7}$$

where $\varphi_k(f_1, f_2)$ denotes the phase frequency. The amplitude and phase can be calculated by:

$$\begin{aligned} \left|B_h^k(f_1, f_2)\right| &= |H_k(f_1)| \, |H_k(f_2)| \, |H_k(f_1 + f_2)|, \\ \varphi_k(f_1, f_2) &= \varphi_k(f_1) + \varphi_k(f_2) - \varphi_k(f_1 + f_2). \end{aligned} \tag{8}$$

We transform the obtained frequency-domain amplitude complex matrix $\left|B_h^k(f_1, f_2)\right|$ into a trainable grayscale map $x_k = grayscale(\left|B_h^k(f_1, f_2)\right|)$ by gray-value processing. The collection of gray map data $\mathcal{X} = \{x_1, ..., x_K\}$ is eventually generated and used for the model construction.

3.3 BISSIAM Network: A Siamese Network Based Contrastive Learning Framework

Contrastive learning normally engages category similarity and feature compression such as VAE to ensure that the extracted features can be more useful for unsupervised learning tasks. The conceptual contrastive learning can be typically instantiated by a Siamese network [40] where two same images are compared in two distinct paths – the

Algorithm 1 Model Training of BISSIAM Network

Input: $K+1$: the number of UAV classes
$Epoch$: the number of training iterations
$Batch$: training batch size
λ: hyper-parameters
lr: learning rate
θ, θ': The parameters of the core network and clone network
$r(t)$: RF signal
$\mathcal{T}$: The set of image augmentation strategies
Output: Network parameters θ, θ'

1: **for** $epoch = 1$ **to** $Epoch$ **do**
2: **for** $i = 1$ **to** $Batch$ **do**
3: $\{t_1, t_2\} \sim \mathcal{T}$ and $y_i \sim Cat(K+1, p = 1/(K+1))$
4: $|B_i(f_1, f_2)| \leftarrow r_i(t)$
5: $x_0^i \leftarrow grayscale(|B_i(f_1, f_2)|)$
6: $x_1^i, x_2^i \leftarrow t_1(x_0^i), t_2(x_0^i)$
7: $z_1^i \leftarrow G_\theta(f_\theta(x_1^i))$ and $v_1^i \leftarrow Q_\theta(z_1^i)$
8: $z_2^i \leftarrow G_{\theta'}(f_{\theta'}(x_2^i))$ and $v_2^i \leftarrow Q_{\theta'}(z_2^i)$
9: $\mathcal{L}_1^i(\theta, \theta') \leftarrow$ Eq.10 with $(v_1^i, z_2^i, v_2^i, z_1^i)$
10: **for** $j = 0$ **to** 2 **do**
11: $y_j^i \leftarrow softmax(v_j^i)$ // obtain the prediction categories
12: **end for**
13: $\mathcal{L}_{CMI}^i \leftarrow \sum_j I(x^i, y_j^i)$ // calculated by Eq. 12
14: **end for**
15: $\mathbb{E}[\mathcal{L}_{\text{LBO}}] \leftarrow \frac{1}{Batch} \sum_{i=1}^{Batch} (\mathcal{L}_1^i(\theta, \theta') - \mathcal{L}_{CMI}^i)$
16: $\theta^* \leftarrow$ Eq. 14
17: $\theta, \theta' \leftarrow \theta^*$ //parameters cloning
18: **end for**
19: **return** Network parameters θ, θ'

image augmentation can facilitate further the key feature extraction of the same class without a focus on other class-independent features. The key idea is to find the similarity of the inputs by comparing the feature vectors of the twin networks and ultimately compute the comparable vector outputs. This section provides technical details of how we construct, train and optimize the BISSIAM network. Alg. 1 describes the pseudo-code of the model construction and training.

3.3.1 Siamese Network Design

Fig. 4 shows the detailed design of the Siamese Network. The network consists of two neural networks – core network and clone network, and we define the weights of the core network and the clone network as θ and θ'. The core network encompasses a feature extractor f_θ (e.g., residual network (ResNet) [41]), a projection multi-layer linear perceptron (MLP) head G_θ and a prediction MLP head Q_θ [38]. The clone network has exactly the same structure as the core network, and its weights are also shared with θ.

We use bispectral grayscale maps $\mathcal{X} = \{x_1, ..., x_k\}$ as the input of the siamese network architecture. First, we get a pair of augmented bispectral views $x_k^1 = t_1(x_k)$ and $x_k^2 = t_2(x_k)$ of the image x_k by applying an image augmentation set $t \sim \mathcal{T}$, such as image cropping or rotation (Alg. 1 Lines 3-6). Then we feed the first augmented bispectral views x_k^1 into the network f_θ and G_θ to get the mapping $z_1 \triangleq G_\theta(f_\theta(x_k^1))$. After further calculation by prediction MLP head Q_θ, we can get the vector encoding $v_1 \triangleq Q_\theta(G_\theta(f_\theta(x_k^1)))$. Similarly, the augmented views x_k^2 are

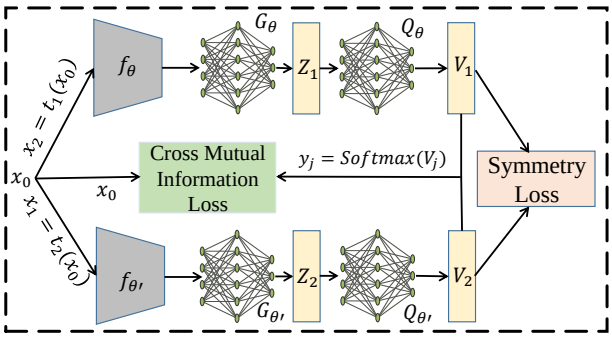


Fig. 4. The design of BISSIAM network based on the siamese network. The network is trained by a combination of symmetry loss and cross mutual information loss, where $y_j (j \in \{0, 1, 2\})$ corresponds to the predicted labels of three images x_0, x_1, x_2 through linear classification layers.

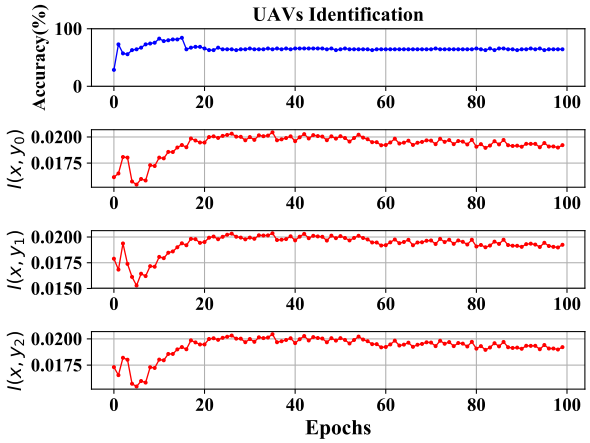


Fig. 5. A high correlation between UAVs recognition accuracy and category mutual information $I(x, y)$ in BISSIAM.

fed into the cloning network and we can ultimately acquire the encoding $v_2 \triangleq Q_{\theta'}(G_{\theta'}(f_{\theta'}(x_k^2)))$ (Alg. 1 Lines 7-8). We swap the output vectors of the two bispectral images to minimize the negative cosine similarity ($Sim.$):

$$Sim.(v_1, z_2) = -\frac{v_1}{\|v_1\|_2} \cdot \frac{z_2}{\|z_2\|_2}, \tag{9}$$

where $\|\bullet\|_2$ is ℓ_2-norm.

Eventually, to identify the pertaining category, we add a classifier layer (e.g., Softmax) between the vector encoding delivered by the siamese network and the category label. Thereafter, we can predict the label, i.e., $Y = \{y_1, y_2, ..., y_{K+1}\} : V \rightarrow Y$, which is similar to [15] (Alg. 1 Lines 10-12). To train the siamese network, we consider both cross mutual information loss and symmetry loss (Alg. 1 Line 9 and Lines 13-17 and more details in §3.3.2).

The learnt model can be then used in the real-world in-sample UAV anomaly detection. To better tweak the prediction effectiveness and reflect the real categorization based on the pre-trained siamese network, we further investigate a small number of labeled samples and associate the estimated category labels Y with their real labels Y_{true}.

3.3.2 Model Training Objectives

Minimizing the symmetry loss. Inspired by [37], our primary training goal is to minimize the symmetry loss of the

bispectrum $\mathcal{L}_1(\theta, \theta')$:

$$\mathcal{L}_1(\theta, \theta') = \frac{1}{2} Sim.(v_1, z_2) + \frac{1}{2} Sim.(v_2, z_1). \tag{10}$$

By learning and comparing the output features, BISSIAM can easily catch the main representation features of the bispectral image, and the resultant representation features (Z and V) can facilitate the execution of the downstream tasks.

Similar to [37], [38], we use the stop-gradient ($stopgrad$) operation and the encoded vector v_1, instead of nonlinear mapper z_1, to prevent the collapse, i.e., $Sim.(v_1, stopgrad(z_2))$. In contrastive learning approaches that only use similar samples, the network learns the degeneracy occurs, i.e., the mapped features $Z = \{z_i\}_{i=1}^k$ of all samples may be fixed as constants. $stopgrad(z_2)$ prevents the gradient quadratic back-propagation of z_2 from an early degradation of the network. The prediction head Q_θ is an average prediction of the mapper feature z – similar to the clustering centroid of *K-means* – that can make the network develop in a correct learning direction [37].

Maximizing the cross category mutual information loss. The cross-category mutual information $I(x, y)$ is referred to as the mutual information value between the predicted category $\{y_0, y_1, y_2\}$ of the original bispectral map x_0 and the augmented images $\{x_1, x_2\}$. As shown in Fig. 5, there is a high correlation between $I(x, y)$ and the prediction accuracy. We can also observe a varying trend of accuracy during the learning process - an initial ramping-up with a decrease before a stabilization, indicating a huge improvement room. To further harvest the accuracy gain, we choose the cross mutual information loss as an additional training objective on the multiple augmented images.

To calculate $I(x, y)$, we assume the prior of $p(y_i)$ for $i \in \{1, 2, ..., K+1\}$ follows the categorical distribution $Cat(K+1, p = 1/(K+1))$ (Alg. 1 Line 3). Thus, we introduce $I(x, y_j)$ for any $j \in \{0, 1, 2\}$:

$$I(x, y_j) = \mathbb{H}[y_j] - \mathbb{H}[y_j|x], \tag{11}$$

where $\mathbb{H}[y_j]$ is the entropy of y_j and $\mathbb{H}[y_j|x]$ is the conditional entropy of y_j given x.

We can then define the cross category mutual information loss as $\mathcal{L}_{CMI} = \sum_{j \in \{0,1,2\}} I(x, y_j)$. Maximizing $I(x, y_j)$, can introduce the influence of the original information x in the pseudo label y_j of the current different augmented views. For $I(x, y_j)$, we derive the variational process as

$$\begin{aligned} I(x, y_j) &= \mathbb{H}[y_j] - \mathbb{E}_{p(x,y_j)}[-\log p(y_j|x)] \\ &= \mathbb{H}[y_j] + \mathbb{E}_{x \sim p(x)}[KL(p(y_j|x)||q(y_j|x))] \\ &\quad + \mathbb{E}_{y_j \sim p(y_j|x), x \sim p(x)}[\log q(y_j|x)] \\ &\geq \mathbb{H}[y_j] + \mathbb{E}_{y_j \sim p(y_j|x), x \sim p(x)}[\log q(y_j|x)] \\ &= \mathbb{H}[y_j] - \mathbb{E}_{x \sim p(x)}[\mathbb{H}[q(y_j|x)]] \end{aligned}, \tag{12}$$

where $KL(\bullet)$ denotes the Kullback–Leibler divergence, and $q(y_j|x)$ denotes the auxiliary distribution for estimating $p(y_j|x)$. The design intention is to ensure the y_j predicted by $q(y_j|x)$ at each time to be a specific class. Hence a low information entropy $\mathbb{E}_{x \sim p(x)}[\mathbb{H}[q(y_j|x)]]$. Then, $\mathbb{H}[y_j]$ can be calculated by $\mathbb{H}[q(y)]$. A larger $\mathbb{H}[q(y)]$ implicates a much balanced distribution of the prediction category rather than a skew categorization, and vice versa.

Putting them together. As the $KL(p(y_j|x)||q(y_j|x))$ is non-negative and the prior-distribution $p(y_j|x)$ is unknown, a lower bound $\mathcal{L}_{LBO}$ can be obtained for the full loss function as follows:

$$\begin{aligned} \min_{\theta, \theta'} \mathcal{L}_{\text{LBO}} &= \min_{\theta, \theta'} \mathcal{L}_1(\theta, \theta') - \mathcal{L}_{CMI} \\ &= \min_{\theta, \theta'} -\tfrac{1}{2} \tfrac{v_1}{\|v_1\|_2} \cdot \tfrac{z_2}{\|z_2\|_2} - \tfrac{1}{2} \tfrac{v_2}{\|v_2\|_2} \cdot \tfrac{z_1}{\|z_1\|_2} \\ &\quad - \sum_{j \in \{0,1,2\}} \mathbb{H}[q(y_j)] - \lambda_j \mathbb{E}_{x \sim p(x)} \mathbb{H}[q(y_j|x)], \end{aligned} \tag{13}$$

where the hyper-parameter λ_j is used for weighting the cross mutual information loss. We use stochastic gradient descent (SGD) to optimize the loss function $\theta^* = \arg\min_{\theta, \theta'} \mathbb{E}[\mathcal{L}_{\text{LBO}}]$:

$$\theta^* \leftarrow SGD(\nabla_{\theta, \theta'}, \mathbb{E}[\mathcal{L}_{\text{LBO}}], \theta, \theta', lr), \tag{14}$$

where $\nabla_{\theta, \theta'}$ and lr denote the gradient and learning rate, respectively.

3.4 p-Sampling based Training Optimization

While the contrastive learning framework proposed in §3.3 can provision an elementary representation, the training efficiency has yet been fully explored. In reality, training with a huge number of samples is time-consuming and resource-intensive, impeding the further integration with real-world surveillance systems.

3.4.1 Core Idea

We aim to ascertain the most suitable proportion of all samples that can holistically balance the accuracy and training efficiency, i.e., selecting a subset of all samples S_p, taking up p proportion of the totality to minimize the overall cost $\mathcal{J}(S_p)$. Presumably the cost can encompass the misclassification (J_1) and the involved cost of computation (J_2), i.e., $\mathcal{J}(S_p) = J_1 + J_2$. Hence, the optimal sample proportion p^* that can minimize the $J(S_p)$ is:

$$p^* = \arg\min_{0 < p < 1} \mathcal{J}(S_p). \tag{15}$$

Essentially, p^* indicates a proportion threshold which can assure good enough accuracy, i.e., making valid predictions with a balanced time-efficiency. Otherwise, if $p \neq p^*$ and $\mathcal{J}(S_p) > \mathcal{J}(S_{p^*})$ for $p < p^* < 1$, Eq. 15 is proved to be not converged.

3.4.2 Cost Breakdown

We firstly use $J_1(y_p, \widetilde{y}_p)$ to define the model misclassification. If $\widetilde{y}_p : r_p(t) \rightarrow \widetilde{y}_p$ denotes the predicted label of UAV signal $r_p(t)$ with sample proportion p, the cost can be expressed as:

$$J_1(\tilde{y}_p, y_p) = 1 - \sum_{i=1}^{N \times p} \delta(y_i, map(\tilde{y}_i))/(N \times p), \tag{16}$$

where N denotes the sample number and $map(\bullet)$ maps the predicted label $\tilde{y}_i$ to the ground truth label. The best mapping can be found through the Kuhn-Munkres (KM) algorithm [42]. δ denotes the delta function is a piecewise function – it equal to 1 when $y_i = map(\tilde{y}_i)$, otherwise 0.

To specify the J_2, we consider two critical metrics – the sample storage cost s_p and the training time cost τ_p. They are integrated into $J_2(S_p) = J_2(s_p, \tau_p)$. s_p can be

Algorithm 2 p-Sampling Algorithm

Input: $\mathcal{D} = \{\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_N\}$:Bispectral grayscale image dataset
$f_\theta, G_\theta, Q_\theta$: Network model
p: Proportion of dataset
$Epoch$: Number of training iterations
ζ: Step
Output: p^*: Optimal dataset proportion
1: **for** $p = 0$ **to** 1 **do**
2: $S_p = \{\mathcal{X}_i\}_{i \in \{1,...,N \times p\}}$ // obtain the sample subset
3: $s_p = p$
4: **for** $epoch = 1$ **to** $Epoch$ **do**
5: **for** $i = 1$ **to** $N \times p$ **do**
6: Update the network $f_\theta, G_\theta, Q_\theta$
7: Calculate time consumption τ_p^i
8: **end for**
9: **if** $epoch > epoch^*$ **then** // await $epoch^*$ until the accuracy is stable
10: $J_1^{epoch} \leftarrow$ Eq. 16
11: $J_2^{epoch} \leftarrow$ Eq. 17
12: **end if**
13: **end for**
14: $J(S_p) = \frac{1}{Epoch - epoch^*} \sum_{epoch^*}^{Epoch} (J_1^{epoch} + J_2^{epoch})$
15: $p \leftarrow p + \zeta$ // update the sample proportion p by step ζ
16: **end for**
17: $p^* = \arg\min_{0<p\leq 1} J(S_p)$
18: **return** p^*

Algorithm 3 Out-of-Sample UAV Detection

Input: $V = \{v_1, v_2, ..., v_N\}$: Node
v_{new}: New UAV node
p^*: Optimal proportion of dataset
Output: y_{new}: The prediction class of v_{new}
1: $V_{p^*} \leftarrow \{v_i\}_{i \in \{1,...,N \times p^*\}}$ // a small number of encoding vectors with a ratio of p^*
2: $G_{p^*}(V_{p^*}, E_{p^*}) \leftarrow$ Eq. 18 // constructing the fingerprint database
3: **for** $j = 1$ **to** $K + 1$ **do**
4: **for** $i = 1$ **to** $N * p^*$ **do**
5: $\mathcal{S}(v_i^j, G_{p^*}^j) \leftarrow$ Eq. 19
6: $\phi_j^* = \min \mathcal{S}(v_i^j, G_{p^*}^j)$ // ϕ_j^* denotes a lower bound similarity for the j-th fingerprint
7: **end for**
8: $\mathcal{S}(v_{new}, G_{p^*}^j) \leftarrow$ Eq. 19
9: **end for**
10: $y_j \leftarrow$ Eq. 20 // most likely category of the new node v_{new}
11: **if** $\min_{j \in \{1,...,K+1\}} \mathcal{S}(v_{new}, G_{p^*}^j) > \phi_j^*$ **then**
12: $y_{new} = y_j$ // Known UAV
13: **else**
14: $y_{new} = y_{new}{}'$ // Unknown UAV
15: **end if**
16: **return** y_{new}

simply estimated by the proportion of samples p given that the storage space will be linearly increased with the increment of samples. We employ $\mathbb{E}[\tau_p]$ i.e., the iteration training time on average, to approximate τ_p. Putting them together, $J_2(s_p, \tau_p)$ can be then calculated by:

$$J_2(s_p, \tau_p) = p + \frac{1}{N \times p} \sum\nolimits_{i=1}^{N \times p} \tau_p^i. \tag{17}$$

Alg. 2 describes the key procedure of the p-sampling mechanism for working out the most suitable sample proportion. We conduct the proportional sampling with the same p onto each categorized data and feed the sampled data into the model training of BISSIAM network.

3.5 Out-of-Sample Detection by Fingerprint Matching

Inspired by [43], we present a new fingerprint matching approach to best identify the UAVs not included in the training procedure. At the core of the rapid detection is to exploit and ascertain the similarity between existing samples and the new target. Alg. 3 outlines the key procedure of the out-of-sample detection. To make it clear, we use v_{new} to generally stand for any new UAV out of the BISSIAM model.

3.5.1 UAV Fingerprint and Sample Distance Measure

We store the vector encodings of all selected samples learnt by BISSIAM representation learning (§3.3) with p-sampling mechanism (§3.4) and use them as the origin fingerprints of the known samples. To indicate the inherent category that a sample tend to pertain to, we use clustering techniques such as K-means to calculate the cluster centroid and approximately categorize each individual sample. To better quantify the relationship between any two samples, we construct an undirected graph $G_p = \{V_p, E_p\}$ as the fingerprint database, where V_p is a collection of in-sample vector encodings (Line 2). The l_2-distance of two node encodings is used to measure the distance between v_i and v_j, and to assign the weight of each edge $e_{i,j} \in E_p$:

$$e_{i,j} = \|v_i - v_j\|_2 . \tag{18}$$

We can form a non-negative adjacency matrix W comprising of $e_{i,j}$. Obviously, a smaller value of $e_{i,j}$ indicates a closer proximity and similarity between two samples; otherwise, two nodes with far distance will have weak similarity and hence high likelihood of different category.

3.5.2 Similarity-Based Matching and Detection

The next step is to calculate the similarity ($\mathcal{S}$) between the encoding vector of a new node v_{new} and any other existing node encodings. To quickly ascertain the closeness to a fingerprint, we calculate the similarity on a per group basis – the average $\mathcal{L}_2$-norm distance and the similarity can be obtained as follow:

$$\mathcal{S}(v_{new}, G_p^j) = 1 - \frac{1}{N(G_p^j)} \sum\nolimits_{i=1}^{N(G_p^j)} \left\| v_{new} - v_i^j \right\|_2, \tag{19}$$

where $N(G_p^j)$ denotes the totality of encoded nodes within G_p^j, the j-th clustering group of the selected samples, and v_i^j represents the individual node encoding. We finally pick up the group with the largest similarity as the most likely category, which has the highest likelihood of analogous behavior and UAV types (Line 10):

$$y_j = \arg\max_{j \in \{1,...K\}} \mathcal{S}(v_{new}, G_p^j). \tag{20}$$

However, simply selecting the maximal similarity cannot fully determine if it is an unknown UAV or not; Instead, we use a *threshold mechanism* to rule out the unknown cases. The threshold can be optimized by quantifying the distances or similarity within a clustering group. Normally we use the satisfactory minimum similarity (lower bound of the similarity ϕ^{min}) associated with the farthest distance (upper

TABLE 2
Network Structure

Projection MLP Head G_θ	**Feature Extractor** f_θ
MLP 1024,ReLU,Batch Norm	ResNet-50
MLP 512,ReLU, Batch Norm	**Prediction MLP Head** Q_θ
MLP 1024, Batch Norm	MLP 512,ReLU,Batch Norm
	MLP 1024

TABLE 3
Configurations

Parameters	**Value**
Step (ζ)	0.02
Hyper-parameter (λ)	[0.5,0.5,0.5]
Number of UAVs (K)	3
Number of UAV operation modes(K')	9
Stabilization iteration point ($epoch^*$)	80
Learning rate (lr)	0.1
SGD Momentum	0.3

bound of the distance γ^{max}) as the required boundaries (Line 6). This is based on an assumption that nodes within a given category typically would not have distances larger than the max distance. We will discuss how we determine the threshold in the real-life problem solving in §5.5. For simplicity, we use the similarity in the algorithm: only if the similarity is greater than a given threshold ϕ_j^*, the predicted label y_j can be entitled to the UAV (Line 12); otherwise the UAV will be regarded as a new intrusion (Line 14).

Furthermore, the fingerprint matching mechanism can be theoretically applied in any categorization. However, a larger K (e.g., operation modes) usually results in the sparse sampling distribution and less samples within each individual clustering class. As a result, a lower categorization effectiveness would manifest. We will discuss this in §5.2.

4 Experimental Setup

4.1 Software and Hardware

The BisSiam detection framework is implemented by using Python 3.8.3 and executed on a server running CentOS 7.3.1611, with Intel(R) Xeon Gold 5118 CPU@2.30GHz and 4 NVIDIA Tesla V100 GPUs. Table 2 depicts the detailed configurations of the networks for the components while the configuration of relevant parameters is shown in Table 3.

4.2 Datasets

We use the dataset [44] in the following experiments. Table 4 details its characteristics where 9 operation modes in total are categorized across three types of UAVs (Parrot Bebop, Parrot AR Drone and DJI Phantom 3). The RF signal data is collected by using National Instruments USRP-2943 (NI-USRP) device with a frequency of 40 MHz for the Wi-Fi radio channel. The main frequency band of the channel is approximately 2.4 GHz and normally no more than 5 GHz. The dataset is 40.3GB with 400 signals and the duration of each signal is 0.25 second while each signal includes 10 million sampling points. It is worth noting that the data labeling usually involve substantial human labor in the loop; most of the labels are delivered though reasoning and intervention in later stages. Therefore, in more general-purpose detection scenarios, the sample labels can be hardly acquired very quickly and precisely.

4.3 Comparative Baselines

We compare our method with several SoTA unsupervised algorithms, including traditional K-means [12] and other deep learning approaches below:

- **VaDE** [13], an auto-encoder based method, that embeds the category information into a VAE [35] by Gaussian mixture model (GMM). We use the result of the auto-encoder as the representation vectors to perform clustering for category prediction.
- **info-GAN** [14], a GAN [34] based method that incorporates latent code mutual information. In this paper, we use the bispectral grayscale map as an adversarial target for an unsupervised classification task via a prior category latent codes.
- **DAAE** [15] that combines AAE [45] with the maximum category mutual information. We use the category latent code of AAE as the classification target.
- **SimCLR** [36] that implements contrastive learning by using siamese networks [40]. We extract the features by comparing two augmented images and perform a classification task on the encoded vectors of the features.
- **SimSiam** [37] that extracts meaningful information and prevents collapsing by using stop-gradient operation. The acquired information will be used for the downstream classification.

4.4 Methodology and Metrics

Methodology. We mainly consider three tasks in the evaluation: UAV presence detection, UAV type detection, and UAV operation mode detection. To construct the desired model, 70% of the dataset is used for training while the remaining for test set. We firstly evaluate the effectiveness of the feature extraction via signal pre-processing (§5.1) and compare the overall effectiveness of prediction among the proposed approach and other baselines (§5.2). To be specific, we evaluate the effectiveness of i) supervised learning approaches given the labels can be fully exploited, and ii) unsupervised learning or semi-supervised learning approaches without the reliance upon a large number of labelled data, which is a more pervasive use case in the real-life UAV detection. We then conduct several micro-benchmarks to examine how different factors such as the parameter number of the model, the number of iterations, etc. on the prediction effectiveness (§5.3). Furthermore, we investigate the benefit from the proposed p-sampling mechanism (§5.4) and examine the effectiveness of the proposed out-of-sample detection (§5.5). Finally the detection time consumption is discussed (§5.6).
Metrics. We use accuracy (Acc), Precision, Recall, F1-score and clustering accuracy (C-Acc) as our evaluation metrics:

$$Acc = \frac{\sum_{k\in K} TP_k + TN_k}{\sum_{k\in K} TP_k + FP_k + FN_k + TN_k}, \tag{21}$$

$$Precision = \frac{\sum_{k\in K} TP_k}{\sum_{k\in K} TP_k + FP_k}, \tag{22}$$

$$Recall = \frac{\sum_{k\in K} TP_k}{\sum_{k} TP_k + FN_k}, \tag{23}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{24}$$

TABLE 4
Dataset Description

Category	Samples	Ratio(%)
No Drone	$820 * 10^6$	18.06
Bebop mode 1: On and connected	$420 * 10^6$	9.25
Bebop mode 2: Hovering	$420 * 10^6$	9.25
Bebop mode 3: Flying	$420 * 10^6$	9.25
Bebop mode 4: Video recording	$420 * 10^6$	9.25
AR mode 1: On and connected	$420 * 10^6$	8.92
AR mode 2: Hovering	$420 * 10^6$	8.92
AR mode 3: Flying	$420 * 10^6$	8.92
AR mode 4: Flying and video recording	$420 * 10^6$	8.92
Phantom mode 1: On and connected	$420 * 10^6$	9.26

$$C - Acc = \sum_{i=1}^{N} \delta(y_i, map(\tilde{y}_i))/N, \tag{25}$$

where TP_k, TN_k, FP_k, FN_k denote the true-positives, true-negatives, false-positives, and false-negatives of k-th category, respectively.

5 Experimental Results

5.1 Effectiveness of Feature Extraction

Observation. Fig. 6 showcases how the RF data, in the form of different categories pertaining to the same UAV (e.g., the Parrot Behop Drone in this example), is extracted and transformed into a trainable format. Observably, the signals are randomly available over time and a substantial amount of data will be aggregated within a short time period, e.g., 10 million data points manifest within 0.25 second, which demonstrates the infeasibility to feed all samples into the model trainer. As shown in Fig. 7, through the procedure of feature extraction depicted in §3.2, the raw RF signals will be transformed into the bispectrum, which can be represented by a bispectral amplitude-frequency matrix $|B(f_1, f_2)|_{128\times128}$ and the equivalent grayscale map. Its dimension is far lower than that of RF data, thereby making it more suitable for the follow-up model training. In addition, the bispectrum can offer more consistent features – two peaks indicate the main band of the signal stays in the 2.4 GHz Wi-Fi band – and overcome the uncertain fluctuation manifesting in the orignal RF data.

Impact on the model accuracy. Apart from the functional validation above, we further investigate the accuracy that the proposed feature extraction can underpin compared against other approaches, i.e., fast Fourier transform (FFT) [22], STFT [26], and RF fingerprint embedding (RFFE) [32]. In particular, the FFT approach converts the values in the frequency domain into grayscale images, while the STFT directly uses time-frequency images for the following detection. The RFFE approach embeds the original RF signal into an unsupervised network info-GAN.

As Signal-to-noise ratio (SNR) is a critical counter that compares the level of desired signal with the level of environment noise, we examine how the noise level impact on the accuracy under different approaches. The varying SNR can be achieved by adding WGN upon a low SNR and we primarily use the method in [21]. More specifically, the SNR of original signal ($SNR_{orignal}$) can be calculated as follow:

$$\begin{aligned} SNR_{orignal} &= 10 \times \log_{10}(\tfrac{P_{signal}}{P_{noise}}) \\ &= 10 \times \log_{10}(\tfrac{\sum_{t=T_e}^{T} |r_k(t)|^2}{T-T_e} / \tfrac{\sum_{t=0}^{T_b} |r_k(t)|^2}{T_b}), \end{aligned} \tag{26}$$

where P_{signal} and P_{noise} represent the power of signal and noise while T_b and T_e denote the start and end point of the signal transient, e.g., the signal emission, respectively. The signal emission typically consumes a huge yet transient energy and hence can be used for specifying the main signal power P_{signal}. Furthermore, we apply WGN ($n(t)$) to the signal to tune the SNR level as below:

$$n(t) = (SNR_{orignal} - SNR_{t\arg et}) \times \mathcal{N}(0, 1), \tag{27}$$

where $SNR_{t\arg et}$ denotes the target SNR.

Fig. 8a and Fig. 8b demonstrate the prediction accuracy comparison among different approaches in the UAV type detection and operation mode detection. There is an obvious ramping-up trend of the accuracy when the SNR soars, i.e., the noise is gradually weakened. This phenomenon is true for every case and BisSiam outperforms all other approaches. Most notably, BisSiam can achieve at least 85% accuracy for the UAV type recognition, even when the SNR is at a noisy level (10dB), indicating the feature extractor in BisSiam can tolerate interference stemming from background noise. Similarly, BisSiam can reach the highest accuracy as opposed to others in the operation mode detection. STFT has the lowest effectiveness simply because the time-frequency images usually experience dynamic changes over time, shown in Fig. 2c, and therefore tend to lower the prediction accuracy inevitably.

5.2 Effectiveness of the In-sample Detection

To evaluate the overall effectiveness of the in-sample anomaly detection, we compare BisSiam with both the supervised and unsupervised learning approaches.

Comparison among supervised learning baselines. Table 5 presents the detailed performance comparison among the comparative supervised models. Apparently, BisSiam outperforms the state-of-the-art methods in terms of $Precision$, $Recall$, and $F1$-score on different tasks. In particular, Table 6 outlines a similar observation of the accuracy (Acc) among different approaches. Most notably, by using BisSiam in supervised learning, the accuracy of detecting the UAV type can reach 98.57% and the accuracy can also retain 92% when detecting the operation mode, which is far higher than other approaches. Additionally, we demonstrate BisSiam can even work effectively when only a fraction of the training set is used. When we cut down the usage proportion to 1/3 (from 100% to 33%), BisSiam can still achieve over 91% accuracy for detecting the UAV types. This indicates its competitive performance and high application potential in resource-constrained training devices such as edge servers or embedding devices. Nevertheless, to more precisely distinguish the specific operation mode, the model still have to involve a larger number of the available sample data.

Comparison among unsupervised learning baselines. As contrastive learning frameworks are based on unsupervised learning, we further compare the performance of our solution with a set of unsupervised learning approaches. To be fair, the same classifier is applied in the UAV detection component of all the comparative baselines. As shown in Table 7, the accuracy of our model is much higher than other baselines, achieving 92.85% and 57.4% when conducting the

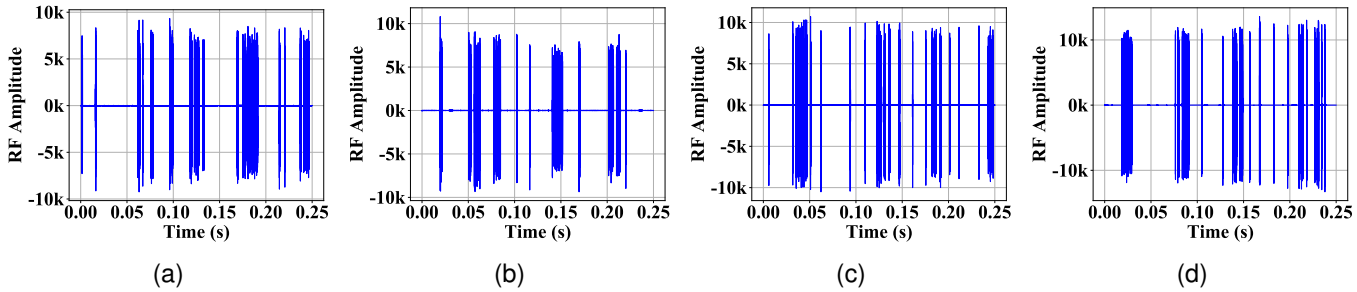


Fig. 6. Parrot Bebop RF signal (a) Mode 1 (*Connected*); (b) Mode 2 (*Hovering*); (c) Mode 3 (*Flying*); (d) Mode 4 (*Video recording*).

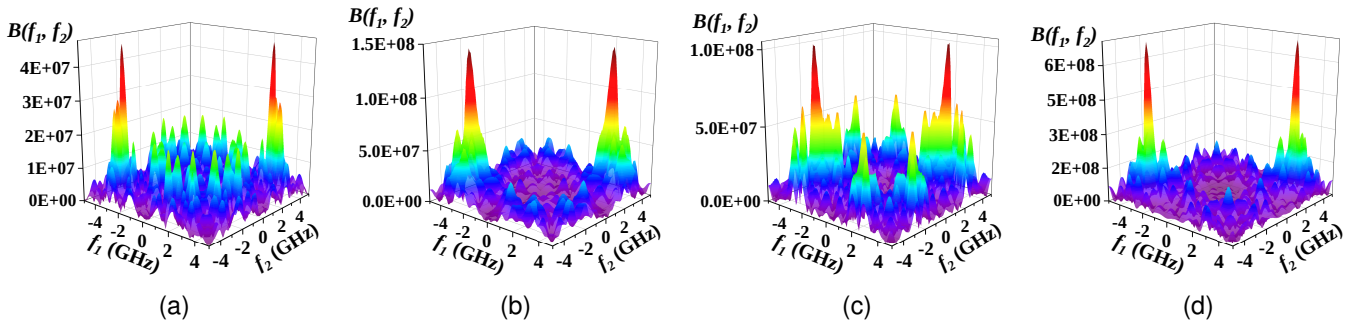


Fig. 7. Parrot Bebop bispectrum: (a) Mode 1 (*Connected*); (b) Mode 2 (*Hovering*); (c) Mode 3 (*Flying*); (d) Mode 4 (*Video recording*).

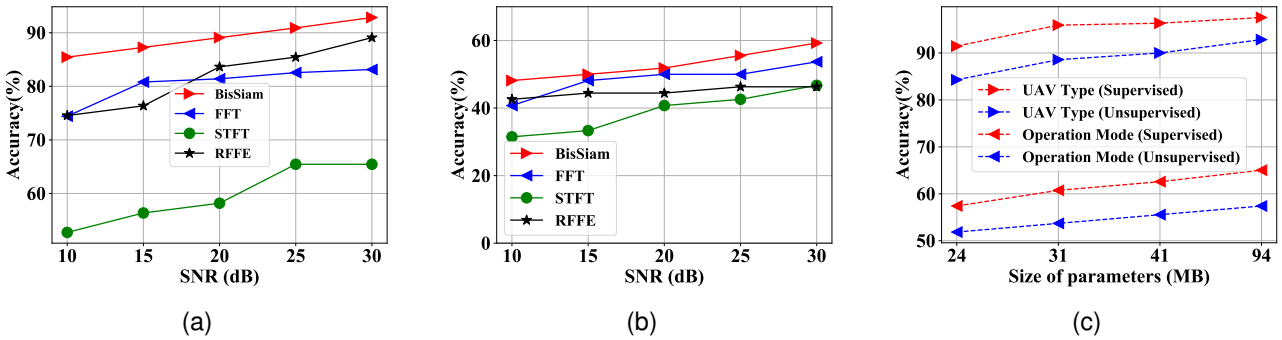


Fig. 8. (a) C-Acc of UAV types (b) C-Acc of operation mode (c) Impact of the parameter size on C-Acc

TABLE 5
Supervised Accuracy Comparison: Precision, Recall and F1

Methods	UAV Presence			UAV Type			Operation Mode		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
DNN [22]	0.996	0.994	0.995	0.910	0.764	0.788	0.535	0.419	0.430
CNN [46]	0.998	0.995	0.997	0.913	0.817	0.846	0.589	0.555	0.551
DRNN [47]	0.999	0.999	0.998	0.908	0.817	0.846	0.589	0.555	0.551
XGBoost [23]	0.999	1.000	1.000	0.900	0.920	0.900	0.650	0.640	0.640
1D-CNN [25]	–	–	1.000	–	–	0.910	–	–	**0.770**
BisSiam (1%labels)	0.980	0.976	0.975	0.939	0.901	0.880	0.948	0.311	0.200
BisSiam (10%labels)	1.000	1.000	1.000	0.958	0.921	0.920	0.9058	0.630	0.533
BisSiam (100%labels)	**1.000**	**1.000**	**1.000**	**0.978**	**0.975**	**0.975**	**0.988**	**0.706**	0.744

task of UAVs type detection and operation mode detection. Among the baselines, the performance of info-GAN is the closest one to BisSiam in the task of type detection. However, the training of GAN network is prone to crashes and thus usually time-consuming. Furthermore, in the task of operation mode detection, DAAE can achieve relatively higher accuracy against other baselines, but it highly relies on the degree of the pre-training, i.e., it is first pre-trained by the reconstruction error of the AE.

Discussion. It is not difficult to ascertain from Table 6 and Table 7 that the supervised learning models tend to be more accurate than the unsupervised learning models. Undoubtedly upfront human intervention is required though and, in most real-life detection scenarios, the human labor is difficult or extremely expensive to acquire. This indicates the necessity of conducting unsupervised learning particularly if the algorithm level solution is integrated with a runtime surveillance system. Where necessary, a compromise would be involving some labeled samples in the model training – e.g., adding the label proportion in Table 6 and Table 7. Consequently, the *semi-supervised* learning can somewhat improve the training accuracy. The second finding is the ac-

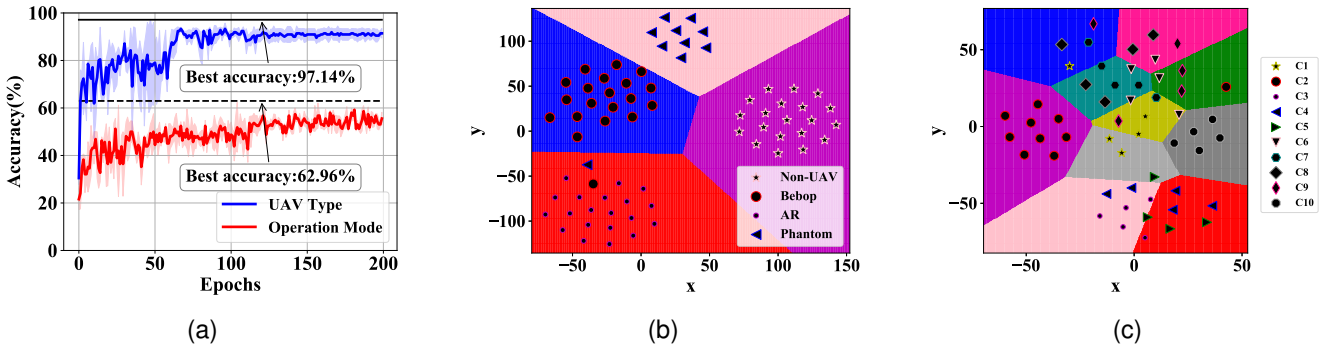


Fig. 9. (a) C-Acc vs. Epochs; (b) UAV type clustering and (c) Operation mode clustering where $C1$-$C10$ represent the modes in Table 4

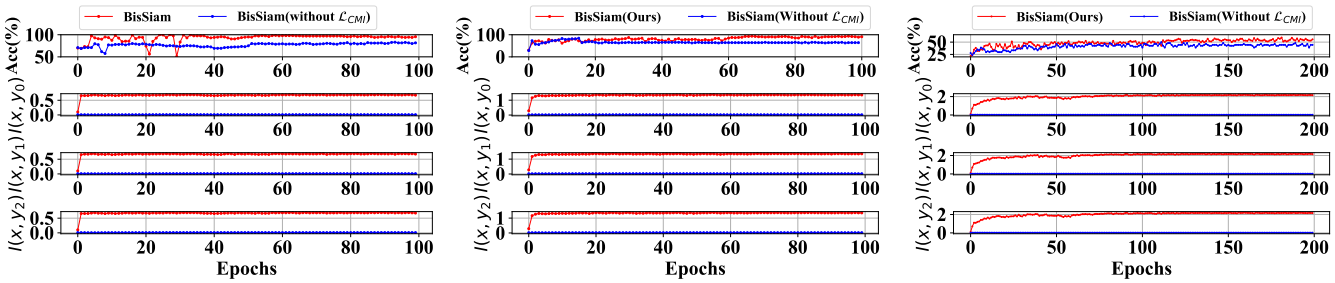


Fig. 10. The C-Acc and the mutual information $I(x, y)$ of the three detection tasks: (a) UAVs presence; (b) UAVs type; (c) UAVs operation mode

TABLE 6
Supervised Accuracy Comparison: Acc

Methods	usage % of training-set	UAV Presence	UAV Type	Operation Mode
DNN [22]	100%	99.70%	84.50%	46.80%
CNN [46]	100%	99.80%	85.80%	59.20%
DRNN [47]	100%	99.90%	90.00%	56.00%
XGBoost [23]	100%	99.96%	90.73%	70.09%
1D-CNN [25]	100%	100%	94.60%	87.40%
BisSiam	**100%**	**100%**	**98.57%**	**92.31%**
BisSiam	**33.3%**	**100%**	**91.72%**	**56.48%**

TABLE 7
Accuracy of the unsupervised/Semi-supervised approaches

Methods	Arch.	Top1 UAV Type	Top1 Operation Mode
BisSiam (supervised)	RN-50	**98.57%**	**68.52%**
K-means [12]	RN-50	48.57%	37.04%
VaDE [13]	AE	75.71%	42.59%
info-GAN [14]	GAN	90.00%	46.30%
DAAE [15]	AE	87.14%	55.55%
SimCLR [36]	RN-50	84.29%	53.70%
SimSiam [37]	RN-50	82.86%	50.00%
BisSiam (unsupervised)	RN-50	**92.85%**	**57.40%**
BisSiam (1%labels)	RN-50	**93.02%**	**59.26%**
BisSiam (10%labels)	RN-50	**95.71%**	**61.11%**

curacy of operation mode detection is far from satisfied and could be improved further. The accuracy results actually derive from the inherent deficiency of the training data that is unevenly-distributed among different categories. Some may only have very few sample data that will directly degrade the detection accuracy. While obtaining more high-quality data would be beneficial to improve the operation mode detection, this requires more effort in tracking and benchmarking more UAVs and collecting their behavioural data. This commitment is currently beyond the scope of this paper and will be left for future work.

5.3 Micro-Benchmarking

In this subsection, we investigate the impact of different factors on the effectiveness of BisSiam.

Impact of the parameter number. Fig. 8c shows the changes of accuracy when the size of model parameters grows. To do so, we multiply the width of a ResNet-50 by a factor $\times 1$, $\times 2$, $\times 4$, and $\times 5$ [39]. Apparently, the accuracy ramps up when the number of model parameter gets larger. For instance, the accuracy of unsupervised learning for UAV type and operation mode detection decreases by 7.56% and 5.55% against the maximum accuracy when the number of parameters decreases to 24MB, respectively. Tuning the parameter size is critical to guarantee the desired accuracy – if the parameter size falls down to 24MB, the accuracy will reduce to merely 84.29% and 51.85% for the two tasks, which are in some cases unacceptable.

Impact of the iterations. Fig. 9a illustrates the accuracy changes during the iterative process. Our approach can ultimately achieve 97.14% (average 92%) and 62.96% (average 57%) for the two tasks respectively. Fig. 9b and 9c visualize the clustering of the encoding results via the *t-SNE* tool [43]. It is worth noting that the UAV type clustering is linearly separable, and the background noise is accurately predicted – our model can achieve 100% accuracy in detecting the presence of UAVs. By contrast, the operation mode detection is less linearly-separable. To improve the accuracy., one can perform the type detection followed by the operation mode amid similar UAVs in a cluster.

Impact of $I(x, y_i)$. Fig. 10 shows the effectiveness harvested from the mutual information $I(x, y_i)$ for the three tasks.

TABLE 8
Accuracy on Different Strategies

	Proj. MLP	Pred. MLP	Acc UAV Type	Acc Operation Mode
Crop	✓	–	81.43%	48.15%
	–	✓	85.71%	53.70%
Rotation	✓	–	91.43%	55.56%
	–	✓	**92.85%**	**57.40%**

Apparently BISSIAM has a higher level of cross-category mutual information $I(x, y)$ due to the maximized $\mathcal{L}_{CMI}$ in the design objective, thereby significantly improving the overall accuracy, particularly of the UAV presence and type detection (Fig. 10a and 10b).

Impact of the image augmentation strategies. We set the image augmentation set as $\mathcal{T} = \{Crop, Rotation\}$: $Crop$ [39] uses low-resolution cropping that covers only a small part of the image with only a small computational cost. and $Rotation = \{0°, 90°, 180°, 270°\}$ denotes the rotation of the image at different angles [16]. As shown in Table 8, using rotation strategy of the images can achieve higher accuracy, i.e., 92.85% and 57.40% for the two tasks, respectively, against the crop strategy. This is because, unlike the general life images, the bispectral image contains sequence information for frequencies (f_1, f_2). If cropped, a loss of sequence information will result in a degraded detection. By contrast, the image rotation will not incur a loss of signal information. Instead, it provides a different perspective to facilitate the computer recognition and key information capture.

Impact of the MLP heads. Table 8 also reveals the results of adopting different types of MLP heads. prediction MLP head (Pred. head) has observably better effectiveness than the Projection MLP head (Proj. head) as the training of the siamese network is guided by regarding the Pred. head as the target. In fact, Pred. head is the average feature prediction of Proj. [37] with more effective classification information similar to the central mean of K-means.

5.4 p-Sampling Performance Evaluation

In this section, we evaluate the impact of sample proportion p on the detection accuracy and ascertain the optimal sample proportion p^*. Fig. 11a shows the accuracy of two detection tasks with different sample sizes and error bands. BISSIAM with loss $\mathcal{L}_{CMI}$ can achieve a higher accuracy performance against other model variants. The accuracy gradually improves with the increment of the sample size but the gain tends to be flattened when p reaches a certain level. Meanwhile, the sample storage cost and training time consumption also climb up when the samples increase. Therefore, ideally, the proposed BISSIAM prefers to adopt a moderate sample proportion that can not only keep the sample size but maintain a competitive model accuracy. Fig. 11b and Fig. 11c show the optimal p^* finally chosen by BISSIAM for the two detection tasks, i.e., $p = 0.31$ for type detection and $p = 0.231$ for operation mode detection. This indicates BISSIAM only uses less than $1/3$ samples but achieves 91.72% and 56.48% detection accuracy.

5.5 Effectiveness of Out-of-sample Detection

Visualization of UAV fingerprints. Fig. 12a demonstrates an instance of the fingerprint database. There are four fingerprints and each individual fingerprint belonging to a specific UAV is depicted with a distinct color: the red, black and green ones represent the Parrot AR Drone, Parrot Bebop and DJI Phantom 3, respectively, while the blue one denotes the background noise.

Intra-class distance and the optimal threshold parameter. The dataset used for the experiment includes three classes: background noise, Parrot Bebop, and Parrot AR Drone, and then set p^* to be 0.31. Fig. 12b shows the cumulative probability density function (CDF) of the intra-class distance for each class. We can find that most of the between-node distance in class 1 (background noise) and class 3 (Parrot AR Drone) stays below 0.2, indicating a strong similarity among different sample nodes and the fingerprint thus has a good concentration without obvious outliers. However, the distances within class 2 (Parrot Bebop) are observably more diverse and patchy – 75% of the between-node distances are no more than 0.306, while there are several groups of outliers since three obvious jumps in the CDF, around the distance 0.58, 0.7, and 0.81.

As discussed in §3.5.2, determining the optimal detection threshold requires the exploration and exploitation of the intra-class distances. As outliers rarely manifest in the class 1 and class 3, i.e., the smoothness in the CDF, it is reasonable to choose the upper distance bound (γ^{max}) and the corresponding lower similarity bound (ϕ^{min}) as the target thresholds. The situation is even more complicated for cases such as class-2 where the outlier or problematic node is the norm rather than the exception. Choosing the maximum distance γ_2^{max} as the threshold of the class 2 will lead to more outliers at edge included and wrongly categorized. To exclude outliers from the similarity calculation whilst including most sample nodes, it is much more rational to reduce the γ to a safe value. As a result, γ_2 is chosen to be 0.306 to cover more than 75% sample nodes while the upper distance bound γ_1^{max} and γ_3^{max} are adopted as the final thresholds. The corresponding similarity value $\phi_i^*, i = 1, 2, 3$ is applied into Alg. 3. Fig. 12c also shows the accuracy when the γ_i varies the accuracy of detecting new UAVs is 85.7% on average. For class 2, choosing a parameter ranging from 0.306 to 0.5 can lead to the best detection accuracy (overall 91.4%) - the range is aligned with the plateau in the CDF where no more outliers will be involved and categorized by the detection algorithm. The accuracy will naturally diminish, once more faulty nodes and outliers are included when the threshold γ_i increases.

The overall accuracy is shown in Fig. 13. Simply using the upper distance bound lead to the 85.7% accuracy of UAV type detection, but it can be improved to 91.4% by adopting the optimal threshold. Even more so, the accuracy improvement of detection new UAVs is more significant – the optimal threshold can result in 90% accuracy, 1.8x increased against using the maximal distance.

Discussion. This indicates the necessity to fine-tune the detection threshold when carrying out realistic detection for the incoming unknown UAVs. However, the current threshold mechanism could lead to a lack of generalization. The focus of this work is to develop a novel unsupervised learning for in-sample detection and will be further enhanced to underpin more robust out-of-sample detection.

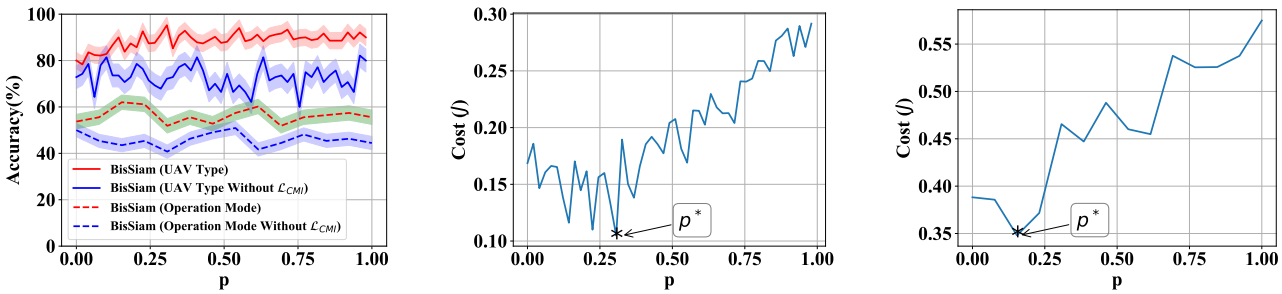


Fig. 11. Sample size evaluation results: (a) Effect of sample proportion on classification accuracy; (b) Cost function curve for drone type recognition; (c) Cost function curve for operational mode recognition.

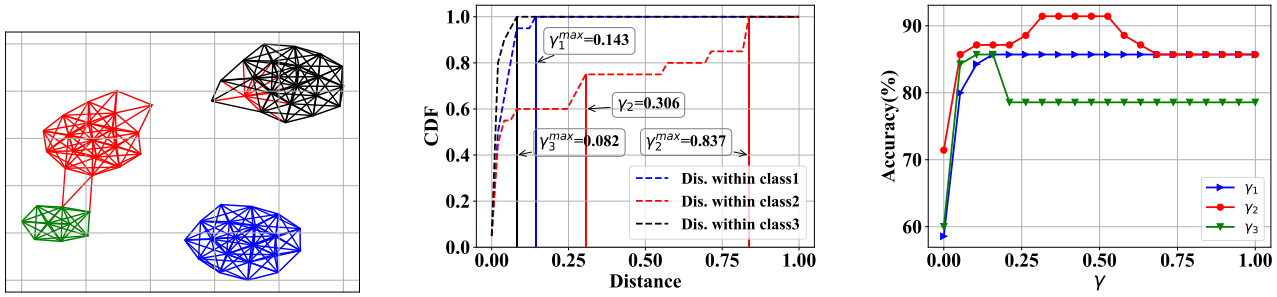


Fig. 12. UAV fingerprint database construction and new UAV detection: (a) Visualization of the fingerprints of UAV types; (b) CDF of the intra-class node distance; (c) Accuracy vs. γ.

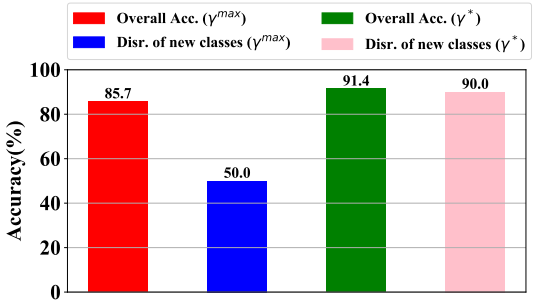


Fig. 13. Accuracy of UAV type and new UAV detection under different threshold strategies.

TABLE 9
Time Consumption

Component	Avg (s)	Stddev (s)
Feature Extraction	1.269	±0.043
BisSiam Prediction	0.053	±0.095
Fingerprint Matching	0.007	±0.006

5.6 Detection Time Evaluation

Table 9 demonstrates the time consumption of a new UAV detection taks and the specific time breakdown into the key components of BISSIAM (outlined in Fig. 3). We mainly measure the average and stddev time of the feature extraction, the vector encoding by BISSIAM network, and the fingerprint matching. Overall, the average time for a detection is 1.329s. On the arrival of a new data sample, the feature extraction takes the dominant proportion of time, 1.269s on average, as opposed to other stages - encoding (0.053s on average) and the final categorization (0.007s on average). The experimental result demonstrate its potential use in a (near) real-time anomaly detection scenarios.

In addition, the evaluation omits the data reading time in the feature extraction; in realistic implementations, this can be implemented in a fairly effective way, for example with the aid of an exclusive hardware device such as USRP for the radio reception. To mitigate the detection delay, BISSIAM can also rely on high-performance hardware to process the massive data or further reduce the sampling ratio.

6 RELATED WORK

Physical signal based UAV detection. Anwar et al. [17] collected the sound signals from the environment and extracted the features by using Mel frequency cepstral coefficients (MFCC) and linear predictive cepstral coefficients (LPCC). The support vector machine (SVM) was then used to classify the UAVs. Seo et al. [18] used a STFT method to transform the UAV sound signal into a spectrogram and used CNN to perform a classification task. Thai et al. [27] used camera to capture the flight video of the UAVs, and employed optical flow to localize and track the flight trajectory of the UAV, through Harris detection and CNN, and finally applied k-nearest neighbor (KNN) for UAV classification. As a departure from the optical-based approaches, Rozantsev et al. [28] stacked the motion windows of UAVs on several consecutive frames and combined them with a regression motion stabilization algorithm to achieve UAV track in low light or almost invisible conditions. However, those sound or optical-based approaches ignore the detectable range and are largely limited by complex environmental conditions such as high levels of noise and poor optical conditions.

Network traffic based UAV detection. Bisio et al. [30] leveraged statistical features of UAV traffic data as the UAV fingerprints to achieve UAV authentication. [29] proposed a ML-based framework for fast UAV detection over the encrypted Wi-Fi communication; it extracted features purely from the packet size and arrival time of encrypted Wi-Fi communication, and designed a delay-aware approach to enable the fast UAV detection with reduced latency. Nevertheless, data privacy became the main concern [26] and it is increasingly difficult to effectively access the target network, as most of the commercial UAVs use proprietary communication channels.

Radio signal based UAV detection. A large body of research employ radar technology for the UAV anomaly detection. Messina et al. [19] sampled the radar received echo (swept frequency) and classified the UAVs with feature extraction by high-pass filter and FFT. Zhang et al. [20] proposed a dual-frequency radar classification scheme where radar data was collected by K-band and X-band radar sensors separately, followed by a STFT transform, and SVM was finally used for the UAV classification. In addition, there are many other work based on radio frequency (RF). [22] released the available UAV RF dataset [44]. It additionally performed Fourier transform analysis of the RF signal and used DNNs for the UAV detection. Medaiyese et al. [23] captured the low frequency spectrum of the RF signals from the UAVs communication with a flight controller. The data was then fed into an Extreme Gradient Boosting (XGBoost) model as the input feature vector. Ezuma et al. [26] extracted RF-based features with the aid of FFT, and, for the first time, used a Markov-based model and a plain Bayesian decision mechanism for detecting RF signals from any source. Zhao et al. [24] improved the auxiliary classifier GANs (AC-GAN) model by leveraging the Wasserstein GANs (WGAN) model. It simplified the recognition steps and can be applied in both indoor and outdoor environments. Ozturk et al. [21] investigated the performance of UAV detection at the low SNR. The models were trained by using both the time-series image and the spectrogram image CNN classifiers, which are illustrated more resilient to the noises. While these RF-based approaches are promising to tackle UAV anomaly detection, the proposed supervised learning models are highly dependent upon massive labelled data. By contrast, this work goes further to investigate an unsupervised learning technique with only a fraction of the whole sample that can mitigate the dependencies on sample sizes whilst retaining a competitive model accuracy.

7 Conclusion

This paper presents BisSiam, a novel learning framework that can transform the UAV radar frequency signals into learnable bispectrum, and learn the vector encoding through a siamese network based contrastive learning model. The vector encodings will be used for the downstreaming detection tasks such as UAV presence, UAV types and operation modes. To achieve a rapid and effective out-of-sample detection, we exploit and ascertain the similarity between existing samples and the new target. A similarity-based fingerprint matching mechanism is devised to detect the unseen UAVs. In the future, we will continue optimizing the accuracy of operation mode prediction and consider the multi-label recognition of multiple UAVs. We also plan to track and benchmark more UAVs and collect their behavioural data as the new datasets.

Acknowledgments

This work is supported by National Natural Science Foundation of China (62072408 and 62073292), Zhejiang Provincial Natural Science Foundation of China (LY20F020030), New Century 151 Talent Project of Zhejiang Province, and UK White Rose University Consortium. This work is also partially supported by UK EPSRC (EP/T01461X/1).

References

[1] V. V. Klemas, "Coastal and environmental remote sensing from unmanned aerial vehicles: An overview," *Journal of Coastal Research*, vol. 31, no. 5, pp. 1260–1267, 2015.

[2] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the sky: Leveraging uavs for disaster management," *IEEE Pervasive Computing*, vol. 16, no. 1, pp. 24–32, 2017.

[3] Y. Huang, S. J. Thomson, W. C. Hoffmann, Y. Lan, and B. K. Fritz, "Development and prospect of unmanned aerial vehicle technologies for agricultural production management," *International Journal of Agricultural and Biological Engineering*, vol. 6, no. 3, pp. 1–10, 2013.

[4] B. Qian, J. Su, Z. Wen, D. N. Jha, Y. Li, Y. Guan, D. Puthal, P. James, R. Yang, A. Y. Zomaya *et al.*, "Orchestrating the development lifecycle of machine learning-based iot applications: A taxonomy and survey," *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1–47, 2020.

[5] H. Peng, R. Yang, Z. Wang, J. Li, L. He, P. Yu, A. Zomaya, and R. Ranjan, "Lime: Low-cost incremental learning for dynamic heterogeneous information networks," *IEEE Transactions on Computers*, 2021.

[6] Y. Hei, R. Yang, H. Peng, L. Wang, X. Xu, J. Liu, H. Liu, J. Xu, and L. Sun, "Hawk: Rapid android malware detection through heterogeneous graph attention networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[7] H. Peng, J. Li, Y. Song, R. Yang, R. Ranjan, P. S. Yu, and L. He, "Streaming social event detection and evolution discovery in heterogeneous information networks," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 15, no. 5, pp. 1–33, 2021.

[8] R. Yang, C. Hu, X. Sun, P. Garraghan, T. Wo, Z. Wen, H. Peng, J. Xu, and C. Li, "Performance-aware speculative resource oversubscription for large-scale clusters," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 7, pp. 1499–1517, 2020.

[9] G. Yeung, D. Borowiec, R. Yang, A. Friday, R. Harper, and P. Garraghan, "Horus: Interference-aware and prediction-based scheduling in deep learning systems," *IEEE Transactions on Parallel and Distributed Systems*, 2021.

[10] Z. Wen, R. Yang, P. Garraghan, T. Lin, J. Xu, and M. Rovatsos, "Fog orchestration for internet of things services," *IEEE Internet Computing*, vol. 21, no. 2, pp. 16–24, 2017.

[11] Z. Wen, T. Lin, R. Yang, S. Ji, R. Ranjan, A. Romanovsky, C. Lin, and J. Xu, "Ga-par: Dependable microservice orchestration framework for geo-distributed clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 1, pp. 129–143, 2019.

[12] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14, 1967, pp. 281–297.

[13] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, "Variational deep embedding: An unsupervised and generative approach to clustering," *ArXiv*, vol. abs/1611.05148, 2016.

[14] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Proceedings of NIPS*, vol. 29, 2016, pp. 2172–2180.

[15] P. Ge, C.-X. Ren, D.-Q. Dai, J. Feng, and S. Yan, "Dual adversarial autoencoders for clustering," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 4, pp. 1417–1424, 2019.

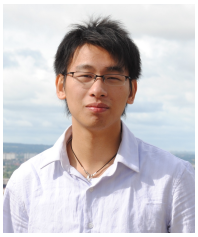
[16] N. Komodakis and S. Gidaris, "Unsupervised representation learning by predicting image rotations," in *Proceedings of ICLR*, 2018.
[17] M. Z. Anwar, Z. Kaleem, and A. Jamalipour, "Machine learning inspired sound-based amateur drone detection for public safety applications," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2526–2534, 2019.
[18] Y. Seo, B. Jang, and S. Im, "Drone detection using convolutional neural networks with acoustic stft features," in *Proceedings of 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2018, pp. 1–6.
[19] M. Messina and G. Pinelli, "Classification of drones with a surveillance radar signal," in *Proceedings of ICCV*, 2019, pp. 723–733.
[20] P. Zhang, L. Yang, G. Chen, and G. Li, "Classification of drones based on micro-doppler signatures with dual-band radar sensors," in *Proceedings of 2017 Progress in Electromagnetics Research Symposium-Fall (PIERS-FALL)*, 2017, pp. 638–643.
[21] E. Ozturk, F. Erden, and I. Guvenc, "Rf-based low-snr classification of uavs using convolutional neural networks," *ArXiv*, vol. abs/2009.05519, 2020.
[22] M. F. Al-Sád, A. Al-Ali, A. Mohamed, T. Khattab, and A. Erbad, "Rf-based drone detection and identification using deep learning approaches: An initiative towards a large open source drone database," *Elsevier Future Generation Computer Systems*, vol. 100, pp. 86–97, 2019.
[23] O. O. Medaiyese, A. Syed, and A. P. Lauf, "Machine learning framework for rf-based drone detection and identification system," *ArXiv*, vol. abs/2003.02656, 2020.
[24] C. Zhao, C. Chen, Z. Cai, M. Shi, X. Du, and M. Guizani, "Classification of small uavs based on auxiliary classifier wasserstein gans," in *Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM)*, 2018.
[25] M. S. Allahham, T. Khattab, and A. Mohamed, "Deep learning for rf-based drone detection and identification: A multi-channel 1-d convolutional neural networks approach," in *Proceedings of 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, 2020, pp. 112–117.
[26] M. Ezuma, F. Erden, C. K. Anjinappa, O. Ozdemir, and I. Guvenc, "Detection and classification of uavs using rf fingerprints in the presence of wi-fi and bluetooth interference," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 60–76, 2019.
[27] V.-P. Thai, W. Zhong, T. Pham *et al.*, "Detection, tracking and classification of aircraft and drones in digital towers using machine learning on motion patterns," in *Proceedings of IEEE ICNS*, 2019.
[28] A. Rozantsev, V. Lepetit, and P. Fua, "Detecting flying objects using a single moving camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 5, pp. 879–892, 2016.
[29] A. Alipour-Fanid, M. Dabaghchian, N. Wang, P. Wang, L. Zhao, and K. Zeng, "Machine learning-based delay-aware uav detection and operation mode identification over encrypted wi-fi traffic," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2346–2360, 2019.
[30] I. Bisio, C. Garibotto, F. Lavagetto, A. Sciarrone, and S. Zappatore, "Unauthorized amateur uav detection based on wifi statistical fingerprint analysis," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 106–111, 2018.
[31] A. Luo, "Drones hijacking," *DEF CON, Paris, France, Tech. Rep*, 2016.
[32] J. Gong, X. Xu, and Y. Lei, "Unsupervised specific emitter identification method using radio-frequency fingerprint embedded infogan," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2898–2913, 2020.
[33] C. Xu, Y. Dai, R. Lin, and S. Wang, "Deep clustering by maximizing mutual information in variational auto-encoder," *Knowledge-Based Systems*, vol. 205, p. 106260, 2020.
[34] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza *et al.*, "Generative adversarial networks," *ArXiv*, vol. abs/1406.2661, 2014.
[35] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *ArXiv*, vol. abs/1312.6114, 2013.
[36] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of ICML*, 2020, pp. 1597–1607.
[37] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proceedings of CVPR*, 2021, pp. 15 750–15 758.
[38] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar *et al.*, "Bootstrap your own latent: A new approach to self-supervised learning," *ArXiv*, vol. abs/2006.07733, 2020.
[39] M. Caron, I. Misra, J. Mairal *et al.*, "Unsupervised learning of visual features by contrasting cluster assignments," in *Proceddings of NIPS*, 2020.
[40] G. Koch, R. Zemel, R. Salakhutdinov *et al.*, "Siamese neural networks for one-shot image recognition," in *Proceedings of ICML deep learning workshop*, vol. 2, 2015.
[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of CVPR*, 2016, pp. 770–778.
[42] L. Lovász and M. D. Plummer, *Matching theory*, 2009, vol. 367.
[43] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of Machine Learning Research*, vol. 9, no. 11, 2008.
[44] M. S. Allahham, M. F. Al-Sád, A. Al-Ali, A. Mohamed, T. Khattab, and A. Erbad, "Dronerf dataset: A dataset of drones for rf-based detection, classification and identification," *Data in Brief*, vol. 26, p. 104313, 2019.
[45] A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow, "Adversarial autoencoders," in *Proceedings of ICLR*, 2016.
[46] S. Al-Emadi and F. Al-Senaid, "Drone detection approach based on radio-frequency using convolutional neural network," in *Proceedings of 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, 2020, pp. 29–34.
[47] A. Gumaei, M. Al-Rakhami, M. M. Hassan, P. Pace, G. Alai, K. Lin, and G. Fortino, "Deep learning and blockchain with edge computing for 5g-enabled drone identification and flight mode detection," *IEEE Network*, vol. 35, no. 1, pp. 94–100, 2021.

Taotao Li is currently working toward the master degree in control theory and control engineering with the College of Information Engineering, Zhejiang University of Technology, Hangzhou, China. His research interests include security and privacy of Internet of Things devices, and data-driven security.

Zhen Hong (Member, IEEE) received the B.S. degree from Zhejiang University of Technology (China) and University of Tasmania (Australia) in 2006, respectively, and the Ph.D. degree from the Zhejiang University of Technology in Jan. 2012. He was an associate professor with the Faculty of Mechanical Engineering & Automation, Zhejiang Sci-Tech University, China. Since Apr. 2019, he is an associate professor with the Institute of Cyberspace Security, and College of Information Engineering, Zhejiang University of Technology, China. He has visited at the Sensorweb Lab, Department of Computer Science, Georgia State University in 2011. He also has been at CAP Research Group, School of Electrical & Computer Engineering, Georgia Institute of Technology as a research scholar in 2016 to 2018. His research interests include cyber-physical systems, Internet of things, wireless sensor networks,cybersecurity, and data analytics. He received the first Zhejiang Provincial Young Scientists Title in 2013 and the Zhejiang Provincial New Century 151 Talent Project (The Third-Level) in 2014. He is a member of IEEE, CCF and senior member of CAA, and serves on the Youth Committee of Chinese Association of Automation and Blockchain Committee and CCF YOCSEF, respectively.

Qianming Cai is currently working toward the master degree in control theory and control engineering with the College of Information Engineering, Zhejiang University of Technology, Hangzhou, China. His research interests include data-driven security and code clone detection.

Li Yu (Member, IEEE) is currently a Professor at College of Information Engineering, Zhejiang University of Technology. He has authored or co-authored three books and over 300 journal papers. His current reserach interests include cyber-physical systems security, networked control systems, motion control and information fusion.



Zhenyu Wen (Member, IEEE) is currently a Tenure-Tracked Professor with the Institute of Cyberspace Security, Zhejiang University of Technology. His current research interests include IoT, crowd sources, AI system, and cloud computing. For his contributions to the area of scalable data management for the Internet of Things, he was awarded the the IEEE TCSC Award for Excellence in Scalable Computing (Early Career Researchers) in 2020.

Renyu Yang (Member, IEEE) is an EPSRC-funded Research Fellow with the University of Leeds, UK. He was with Beijing Advanced Innovation Center for Big Data and Brain Computing, China, Alibaba Group China and Edgetic Ltd. UK, having industrial experience in building large-scale distributed systems with ML and co-authored/co-led many research grants including UK EPSRC, Innovate UK, EU Horizon 2020, China 973/863, etc. His research interests include distributed systems, resource management and applied machine learning.