

## Article

# Learning-Based Anomaly Detection and Monitoring for Swarm Drone Flights

Hyojung Ahn <sup>1,2</sup>, Han-Lim Choi <sup>2,\*</sup> , Mingu Kang <sup>3</sup> and SungTae Moon <sup>1,2</sup><sup>1</sup> Korea Aerospace Research Institute, Daejeon 34133, Korea; hjahn@kari.re.kr (H.A.); stmoon@kari.re.kr (S.M.)<sup>2</sup> Department of Aerospace Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, Korea<sup>3</sup> School of Mechanical Engineering, Pusan National University, Busan 46241, Korea; mgkang1994@pusan.ac.kr

\* Correspondence: hanlimc@kaist.ac.kr

Received: 22 October 2019; Accepted: 11 December 2019; Published: 13 December 2019



**Abstract:** This paper addresses anomaly detection and monitoring for swarm drone flights. While the current practice of swarm flight typically relies on the operator's naked eyes to monitor health of the multiple vehicles, this work proposes a machine learning-based framework to enable detection of abnormal behavior of a large number of flying drones on the fly. The method works in two steps: a sequence of two unsupervised learning procedures reduces the dimensionality of the real flight test data and labels them as normal and abnormal cases; then, a deep neural network classifier with one-dimensional convolution layers followed by fully connected multi-layer perceptron extracts the associated features and distinguishes the anomaly from normal conditions. The proposed anomaly detection scheme is validated on the real flight test data, highlighting its capability of online implementation.

**Keywords:** swarm drone; anomaly detection; clustering; labeling; classification

## 1. Introduction

Coordination of multiple drones has been intensively studied in the field of military surveillance, smart agriculture, logistics, disaster response, and artistic drone shows [1,2]. The use of swarm of small aerial vehicles allows for extended mission area, flexible mission capability, robustness to single point failure, and cost effectiveness. Research areas related to swarm drones have been very diverse, including development of small-scale aerial vehicles [3–5], ad hoc communication backbone tailored to swarm operation [6–8], path generation to ensure collision avoidance [9–11], mission-level planning and scheduling to achieve high-level autonomy [12–15], and interaction/interface between the human operator and the swarm drones [16–19]. It should be noted that while earlier literature focused on technologies to enhance performance and capabilities, recent work has been looking into more safe, secure, and reliable operations of such swarm systems [20–23]. In order to operate in a more advanced, robust, scalable and flexible manner, research with self-regulation and environmental adaptation is expanding into the design of new coordination algorithms that combine biological processes [24].

For the safe and reliable operation of swarm drones, it is crucial to be able to monitor/manage the health of the vehicles and to take prognostic actions when needed. The system health may be affected by many aspects such as faults in the actuator and sensors in the drone system, defects in the communication link, and possibly cyberattacks by malicious entities; identification of the culprit requires incorporation of detailed knowledge of the system, mission, and the environment. That said, the first step toward health management is to detect an anomaly in the system behavior—in other words, to recognize something goes wrong (even without clear identification of what went wrong).

Unfortunately, the current practice of health monitoring for swarm drones often relies solely on the human operator's experience and expertise—eyeballing flying drones' motion and/or looking at the trajectory data on the ground station screen. Deeper post-flight investigation can certainly be performed, but on-site monitoring of vehicle health is critical to enable execution of an appropriate contingency plan and to avoid secondary damage due to a chain reaction of failures. As such, it is imperative to devise an automated procedure to monitor and detect an anomaly of flying swarm drones to support the operator's correct situational awareness and responsive decision making.

Traditional model-based methods have been intensively studied in the context of fault detection and isolation (FDI) particularly for safety-critical aerospace systems [25,26]. This has recently been extended to multi-unmanned aerial vehicle (UAV) systems from the perspective of safe operations [27–29] and also for resilience against cyberattacks [30–33]. In a multi-UAV system, the design and experimental verification methods and strategies have been studied to perform fault detection, identification, and recovery (FDIR) by constructing a cooperative virtual sensor (CVS) system for each UAV through on-board sensor signals [34]. These model-based schemes work well when normal behavior of the system is well-understood and predicted using (physics-based) models and also when the potential fault modes are well-identified a priori so that a finite number of hypotheses on the faultiness can be posed. However, this is not often the case for a flight of swarm drones. From the mission perspective, one reason to utilize the swarm is their robustness to single-point failure; however, it is still critical to ensure the safety and health of each vehicle that may affect the overall performance. In particular, a least thus far, small-sized drones are not designed under a high-reliability requirement; thus, the main parameters associated with motion models of the drones are prone to uncertainties. Also, these drones are relatively new in the market, compared to long-life large aircraft; as such, the fault modes are not well understood and identified. Therefore, a data-driven scheme that does not explicitly rely on the physics-based model of the vehicles can be a promising solution to anomaly detection of swarm of small-sized drones.

Several noticeable studies have been performed to take advantage of data-driven machine-learning techniques for fault and/or anomaly detection of aerial vehicles. A multivariate Gaussian mixture model (mGMM) was proposed to detect and monitor airborne abnormalities in aircraft systems in real time [35], and a recurrent neural networks (RNN) method was proposed to identify events and trends that may reduce safety margins of the system out of the data collected from the flight data recorder (FDR) and/or flight operational quality assurance (FOQA) data [36]. A K nearest neighbor (KNN)-based method was presented to estimate the cause of failure and potential degradation of UAV performance on the fly [37]. Research carried out kernel principal component analysis (KPCA)-based sensor anomaly detection using drone sensor signals [38]. It should be noted that these previous studies mostly dealt with health management of a single UAV; there has not been a systematic machine learning-based methodology proposed to effectively detect anomaly in swarm flights, addressing the aforementioned difficulty in behavior of multiple vehicles.

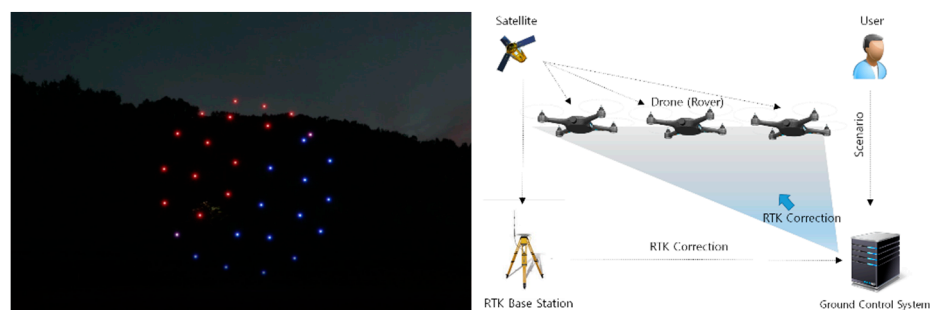
As such, this work presents a learning-based data-driven approach for anomaly detection in swarm flights. The approach primarily tries to tackle the issue of a lack of labeled data in swarm flights, and imbalance in the normal and abnormal data. A sequence of unsupervised learning methods is first applied: a principal component analysis (PCA) [39] procedure first reduces the dimensionality of the time series of flight data, and a K-means clustering groups the data and label them into four categories—true anomaly, uncertain anomaly, uncertain normal, and true normal. Then, a deep neural network, which is concatenation of a one-dimensional convolutional neural network (1D-CNN) [40] and a multi-layer perceptron for logistic regression, is trained to extract features and to detect anomaly. In general, supervised learning methods have better classification performance than unsupervised methods; a very recent form of self-supervised learning that outperforms the fully supervised method is reported in the context of anomaly detection [41]. While the authors of [41] devised highly sophisticated self-supervised framework that performs best, they also pointed out that the fully supervised method ranks the second by dominating all other unsupervised methods.

The present paper is in line with this observation that the supervised learning approach is still a viable top-performing framework if a sufficient amount of labeled data is available and/or appropriate labeling procedure can be accompanied. The proposed method in this paper is tested against real flight data to demonstrate its accuracy in detection of anomaly and also applicability as an online monitoring scheme. The main contributions of this paper are: (a) to propose a systematic procedure of data-driven anomaly detection for swarm flights, and (b) to validate the method against real flight test data.

## 2. Problem Description

Anomaly detection (AD) for swarm flights considered in this work involves two decisions: (a) to detect symptom of anomaly of the overall swarm, and (b) to identify which particular drone behaves in an abnormal manner. The anomaly detection scheme makes these decisions by monitoring the time series of the kinematic variables (i.e., position and velocity) of the drones. In particular, this AD scheme lies in the ground station that manages the overall health and flight status of the swarm vehicles; in other words, if implemented in practice, transmitted data from the drones down to the ground station go through the AD scheme to indicate the normality of the flights.

This work aims at developing such an AD scheme by learning from the data collected in real flight tests. The swarm flight data on which this work is based was collected via a series of flight tests of swarm drones; this swarm system features the use of real-time kinematic (RTK) global positioning system (GPS)-based precision navigation methods proposed in [42]. The flight test data utilized in this work were obtained from a series of swarm flights completed between 26 April 2018 and 22 June 2018; as many as 30 quadcopter drones were used in these flight tests. About 60 tests were conducted, including individual and several groups of test vehicles. From these test results, data were selected that could be used as learning data suitable for the purpose of detecting abnormal behavior during the performance of a swarm mission. Figure 1 illustrates an example mission of the swarm drones (left) and the RTK-GPS navigation architecture of the swarm system (right).



**Figure 1.** Swarm-drones flight-test and system configuration diagram.

The output data from each flight sortie consists of 248 parameters, and the data are in the form of time series of these parameters some of which include multiple measurements. In this study, the following parameters that are thought to be crucial in monitoring anomalies in motion are chosen: three position coordinates and their set point values, three velocity components of the vehicles, and the vehicle status. The position coordinates  $(x_a, y_a, z_a)$  and the velocity components in three axes are obtained from RTK-GPS measurements whose accuracy was verified in [42]; the set point values  $(x_s, y_s, z_s)$  in three coordinates generated from the mission controller are also considered as anomalies that are likely to be associated with the errors between the desired and the actual behavior in vehicle motion, i.e., the following three terms:

$$x = x_a - x_s, y = y_a - y_s, z = z_a - z_s \quad (1)$$

Accelerometer and gyro reads obtained from the inertial navigation system (INS) are also considered as mechanical defects in drone systems that may typically yield abrupt and unpredictable

behavior in acceleration and angular rates. The vehicle status parameter is included for basic checks for data compatibility and calibration. It should be noted that this work tries to devise an anomaly-detection scheme that is not dependent on the specific type and characteristics of a particular drone; as such, the data is not labeled with vehicle IDs. The characteristics of the parameters used in this study are tabulated in Table 1. The vehicle status variables are used to extract data that are collected during the flight when the navigation state is not “manual” and the arming state is “armed” indicating every device in the drone is fully powered. The data is collected with a 10 Hz rate.

**Table 1.** Extracted parameters for flight analysis.

Parts	Variable	Attribute Type
RTK–GPS	Inertial Position and Velocity	Continuous
INS	Gyro (Body frame rate and their integrals)	
Setpoint	Accelerometer ( $x, y, z$ -axis values and integrals)	
Vehicle status	Position ( $x_s, y_s, z_s$ )	Discrete
	Navigation state	
	Arming state	

It should be pointed out that these kinematic variables may not be the complete and comprehensive set of variables needed to detect all possible anomalies in swarm flights. However, these are certainly critical variables in detecting anomalies caused by some types of failures and faults. Thus, this work expresses the anomaly detection problem as the detection and identification of abnormal behavior on the basis of kinematic variables, and focuses on such a problem. Also, this particular flight data is partially labeled, meaning that some of the faulty events identified during the test are identified as anomaly. However, the method in this work does not take into account these pre-identified labels as they are not comprehensive.

### 3. Preliminaries

This section summarizes four machine learning algorithms that are primarily utilized to develop a data-driven anomaly detection scheme in this work. The first two are categorized as unsupervised learning that are devised to handle unlabeled data, and the last two are supervised learning algorithms that take advantage of labeled data for classification tasks.

#### 3.1. Principal Component Analysis (PCA)

Principal component analysis (PCA) is a multivariate statistical projection technique applied for dimensionality reduction [39]. The key idea of PCA is to represent the statistical distribution of data vector using the basis vectors, defined by the eigenvectors of the sample covariance matrix. Let the zero-mean data matrix  $S$  be defined as the row concatenation of  $n$  samples of  $d$ -dimensional row data vectors shifted as zero-mean. Then, the sample covariance matrix is computed as  $\Sigma = S^T S$ . The PCA learns the transform:

$$T_q = S W_q \quad (2)$$

where  $W_q$  is the column concatenation of the  $q$  largest eigenvectors of  $\Sigma$ . Each column of  $T_q$  is called the score vector of the  $d$ -dimensional data vector. The PCA is known to be robust against noises and frequently utilized as a feature extraction method [43].

### 3.2. K-Means Clustering

K-means clustering [44] is a well-established technique to group similar data together. This algorithm finds out the clusters and binds the data to the closest cluster. Mathematically, the clustering is conducted by solving the following optimization:

$$\min_{\{r_{nk}\}, \{\mu_k\}} J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|X_n - \mu_k\|^2 \quad (3)$$

where  $N$  denotes the number of data;  $k$  denotes the number of clusters;  $\mu_k$  denotes the center of the  $k$ -th cluster;  $r_{nk}$  becomes 1 if the  $n$ -th data belongs to the  $k$ -th cluster and 0 otherwise. The K-means algorithm works in two steps. First, a random value is given to  $\mu_k$  as an initial value. In the expectation (E) stage,  $r_{nk}$  is set to minimize  $J$  while fixing  $\mu_k$ . Moreover,  $r_{nk}$  allocates all the data,  $n$ , to the cluster with the smallest distance among all the clusters. In the maximization (M) stage, the newly obtained  $r_{nk}$  is fixed and  $\mu_k$  calculates the mean of the  $k$ -th cluster. The calculation is repeated until the two values converge within an appropriate range. The two steps together are called the expectation–maximization (EM) algorithm.

### 3.3. One-Dimensional Convolutional Neural Network (1D-CNN)

Convolutional neural networks (CNN) have been developed mostly in the context of classification. The basic concept of CNN is to take advantage of locality in the process of feature extraction. While many advancements in CNN has been made to deal with two-dimensional image data, the same concept of locality/sparsity can be equivalently applied in handling one-dimensional time-series data [40]. The network layout of 1D-CNN is similar to the 2D counterpart; it consists of a stack of one-dimensional convolution and max-pooling layers at the end of which is connected a global-pooling layer or a flattened layer.

Often, another fully connected layers (i.e., multi-layer perceptron) are connected to the output layer of the 1D-CNN to perform classification or regression [45]; then, standard learning schemes such as stochastic gradient descent methods can be used to optimize the weights of the CNN and the MLP simultaneously. Note that the classification model in this work also takes this type of structure.

### 3.4. Logistic Regression

Logistic regression (LR) [46] is one of the simplest and commonly utilized machine learning algorithms for two-class classification. LR is a statistical method for predicting binary classes whose results or target variables are inherently dichotomous—say, one and zero. The hypothesis of logistic regression can be written as

$$y = 1/(1 + \exp(-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_m X_m))) \quad (4)$$

where  $X_m$  is the explanatory variables (or features), and  $\beta_i$ 's are associated coefficients to be optimized through a learning process. As the range of  $y$  is between 0 and 1; this output can be interpreted as the probability of belonging to class 1.

The typical loss function to optimize the coefficients is to maximize the likelihood of the output, which takes the form of:

$$\text{loss}(z, y) = - \sum_{i=1}^n (z_i \log y_i + (1 - z_i) \log(1 - y_i)) \quad (5)$$

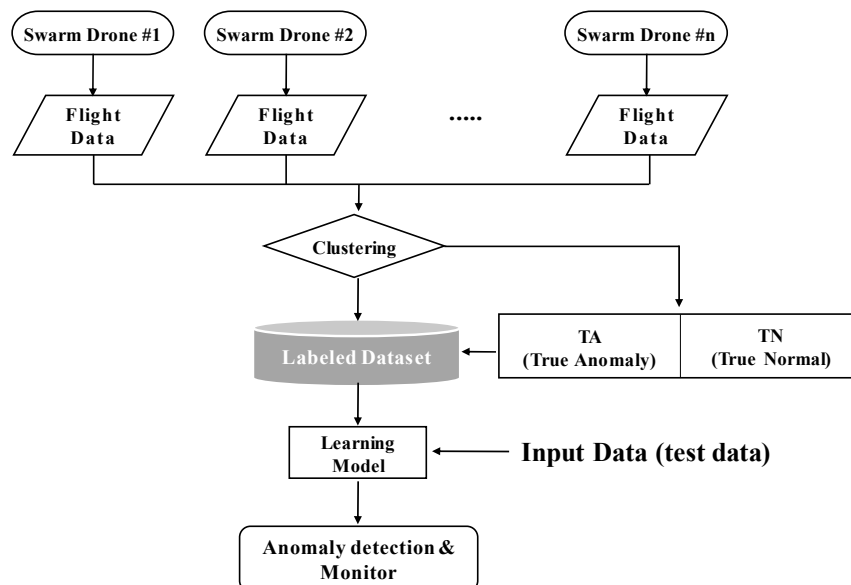
where  $n$  is the number of data and  $z$  is the target value. It can be shown that this loss can also be interpreted as cross-entropy between  $z$  and  $y$ .

The logistic regression can be done in a stand-alone manner, but it is also common to have this sigmoidal activation function at the output not of a neural network to perform classification. In this case, the network is learned to minimize the cross-entropy loss. This paper also takes this approach.

## 4. Learning-Based Anomaly Detection

### 4.1. Model Concept

The overall architecture of the anomaly detection model in this work is illustrated in Figure 2. It is assumed that the flight test data, which may be collected in advance and/or being fed in a real-time manner, are not necessarily labeled. This is common in anomaly detection cases as the data can be clearly labeled only if there exists a working anomaly detector or if human investigators have looked into the data carefully. Therefore, a clustering algorithm is first used to group the data and to label them into several meaningful categories. In this work, given the uncertainty in decisions on anomaly, four categories are considered in the labeling procedure: true normal (TN), true anomaly (TA), uncertain normal (UN), and uncertain anomaly (UA). The latter two categories are introduced as there are cases in which it is not clear whether it is possible to assess if the flight behavior is normal/abnormal or not with solely the given data. Once the labeled data is secured, a 1D-CNN-based binary classifier is trained to learn the key features in determining the anomaly in the data using a set of training data. When the training is completed, the learned model is tested and verified with a separate set of data reserved for this purpose. Then, the learned model can be used as an anomaly detection scheme for both the post-flight analysis and on-line anomaly monitoring. We used the python language (version 3.5, Python Software Foundation, Wilmington, DE, USA) and the open-source library tensorflow (version 1.14.0, Google Brain, Mountain view, CA, USA) framework to implement this deep-learning network.



**Figure 2.** Anomaly detection model for swarm-drone system.

### 4.2. Data Preprocessing

Since the data preprocessing process directly affects the analysis results, it is important to obtain data that is refined in order to develop good algorithms, since the correct analysis results require the correct data to be entered [47,48]. Data preprocessing consists of data cleaning, integration and transformation, and a total of 84,850 time series are obtained as the outcome of this pre-processing.

#### (1) Data Cleaning

In this process, data is refined by filling missing values, filtering excessive noises, removing outliers, and resolving inconsistencies. As explained in Section 2, the difference between the actual position and the reference position, and the actual velocity are used as input data. To match the time stamp of the two different data sources, a linear interpolation technique is used. A linear interpolation of reference data, which is relatively low noise compared to the actual signals, is



performed to derive a difference value from the actual data at the same time stamp. Excessively large values that may be caused by noise aberrations are removed from the training data set. In addition, some portion of data exhibiting noises are cut out to reduce noise values across the data.

(2) Data Integration

Data from multiple drones are integrated into a single dataset with vehicle IDs. The vehicle ID is not explicitly used in the learning process but is useful in analyzing the performance of the learned model. Then, the input data includes 1 label information, 1 time stamp, and 6 kinematic variables.

(3) Data Transformation

Normalization is performed to bring the range of values of the selected parameters within a certain level. The well-known standardization method is used in this paper. The standardization method normalizes each data sequence by calculating the mean value ( $X_{mean}$ ) and the standard deviation value ( $X_{stdev}$ ):

$$X_{standard} = (X - X_{mean})/X_{stdev} \quad (6)$$

### 4.3. Clustering and Labeling

Sensor data are often multi-dimensional and exhibit complicated correlation, making it difficult to characterize the system's state solely on the basis of this data. A principal component analysis (PCA) is utilized for appropriate dimensionality reduction of the data by resolving the major correlation structure between the variables. The underlying rationale of PCA is that most system states can be sufficiently well-represented by the behavior of a few principal components [39]; PCA has been an effective feature extraction scheme in many contexts. In this work, the six-dimensional data representing the kinematic information is represented in terms of principal component axes with a rate of cumulative dispersion of at least 90% using PCA.

This work utilizes clustering to facilitate labeling of the data that are not labeled in their raw form. Labeling allows for implementation of a supervised learning method in anomaly classification. The clustering procedure provides a grouping of data in terms of some type of similarity and distance metric so that human experts are readily able to label the data. It should be noted that the clustering result does not directly lead to classification; the role of clustering is rather limited as an aiding tool for the processing of the data in this work. Specifically, *K*-means clustering on the reduced space defined by the principal component analysis method is proposed in this work. The time-series data that we used for PCA comprised six variables ( $x, y, z; v_x, v_y, v_z$ ) and attached one drone label displayed after PCA. Since the principal component axis is selected so that the cumulative dispersion ratio is 90% or more, the number of dimensions reduced as a result of PCA may be different for each flight data. Clustering is performed using dimension reduced data, and the number of clusters is optimized with the elbow method [49]. By checking the clustered results, the status of each drone in the group is labeled into normal (TN) and abnormal (TA). In other words, the related variables were extracted by dividing into two cases: (i) check the results of performing a given scenario as a mission (RTK-GPS; setpoint data) or (ii) check the sensor signals to detect any anomaly in the vehicle (INS data). These processes including PCA and clustering label the unlabeled data to acquire a labeled data set for learning. As a result, unlabeled dataset can be labeled with two categories, TN and TA. The numbers of data samples in each category are 34,612 (TN) and 31,377 (TA).

### 4.4. Classification

(1) Standardization

After labeling the normal and abnormal state, we removed the drone-label information and used a total of six variables. The range of input variable values are normalized by a standardization method similar to the pre-processing for clustering.

Logistical regression algorithms can be used for binary-categorized variables such as the data labeled in this study. The two possible dependent-variable values, represented by 0 and 1,

correspond to the “normal” and “abnormal” results. Binary logistical models are used to estimate the probability of a binary response based on one or more predictor (or independent) functions.

(2) 1D-CNN Classifier

The overall architecture of the classifier for anomaly detection is illustrated in Figure 3. The time series data of kinematic variables described in Section 2 is first standardized with the procedure described earlier in this section. The data first passes through 1D-CNN consisting of six hidden layers; at the output end of CNN is connected to dense multi-layer perceptron with sigmoid activation function. The output of this sigmoid activation function is compared with the target label value, and their cross-entropy error as explained in (5) is used to learn the overall neural network by propagating the error backwards. A stochastic gradient method with minibatch is used for learning. Table 2 represents more detailed attributes of the neural network layers, and also parameters used to learn the network. The total number of parameters in the training process is 114,198, which is 446 KB in size. The architecture laid out in Table 2 is determined by at first building a sufficiently large network and then implementing batch normalization [50] to regulate the network.

(3) Mini-batch sampling in Adaptive moment estimation (Adam) optimization

Most neural network algorithms work best when learning with the same (or similar) amount of data for each class because most algorithms are designed to maximize accuracy and reduce errors. However, if the number of normal and abnormal individuals is considerably different such as in defect classification or abnormal detection, binary classification in such a class-imbalanced circumstance does not produce good results [51]. To mitigate this imbalance, this work generates additional samples to achieve one-to-one ratio of normal and abnormal data when creating batches for Adam [52] optimization adopted for training of the network.

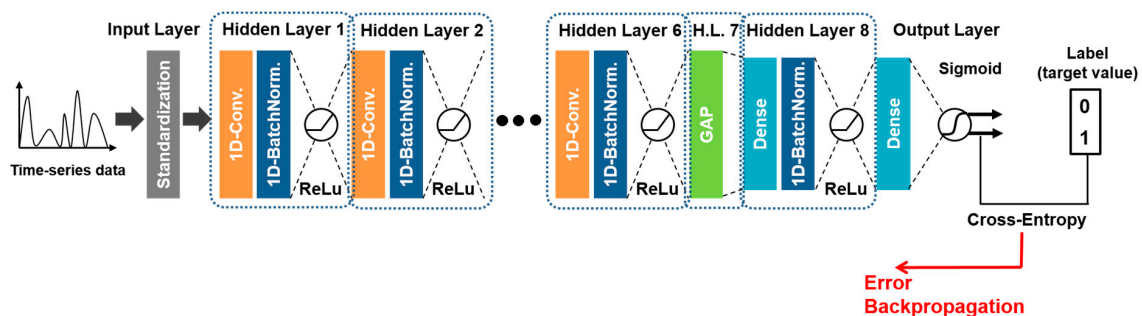


Figure 3. Neural network-based anomaly detection structure.

Let denote the number of data labeled as normal (TN) and abnormal (TA) as  $m_n$  and  $m_a$ , respectively. If sampling to construct a minibatch of size  $b$  in the stochastic gradient calculation is done in a uniform way, the expected ratio of the number of data of the two classes will be  $m_a/m_n$ , which can be very small as an anomaly, as it is rare. Therefore, this work suggests copying the abnormal data set  $k$  times and then sample from this expanded set of data. This way the expected ratio between the two classes can be increased to  $km_a/m_n$ . By appropriately choosing  $k$ , the effect of imbalance can be mitigated. Numerical results in Section 5.2 demonstrate the effectiveness of this sampling scheme.



**Table 2.** Neural network parameters and learning parameters ( $n$ : batch size).

Layer	Input	Output	Operation
Input	$(n, \text{timestamp}, 6)$	$(n, 160, 6)$	Standardization
Hidden 1	$(n, 160, 6)$	$(n, 80, 12)$	1D Convolution (strides 2, kernel size 3), 1D Batch normalization, Rectified Linear Unit (ReLU) [53]
Hidden 2	$(n, 80, 12)$	$(n, 40, 24)$	1D Convolution (strides 2, kernel size 3), 1D Batch normalization, ReLU
Hidden 3	$(n, 40, 24)$	$(n, 20, 48)$	1D Convolution (strides 2, kernel size 3), 1D Batch normalization, ReLU
Hidden 4	$(n, 20, 48)$	$(n, 10, 96)$	1D Convolution (strides 2, kernel size 3), 1D Batch normalization, ReLU
Hidden 5	$(n, 10, 96)$	$(n, 5, 128)$	1D Convolution (strides 2, kernel size 3), 1D Batch normalization, ReLU
Hidden 6	$(n, 5, 128)$	$(n, 5, 128)$	1D Convolution (strides 1, kernel size 3), 1D Batch normalization, ReLU
Hidden 7	$(n, 5, 128)$	$(n, 128)$	Global Average Pooling (GAP) [54]
Hidden 8	$(n, 128)$	$(n, 64)$	Dense, Batch normalization, ReLU
Output	$(n, 64)$	$(n, 2)$	Dense, Sigmoid
Hyper-parameters for learning model			
Batch size 128	Time length 160	Epoch number 50	Learning rate 0.00004

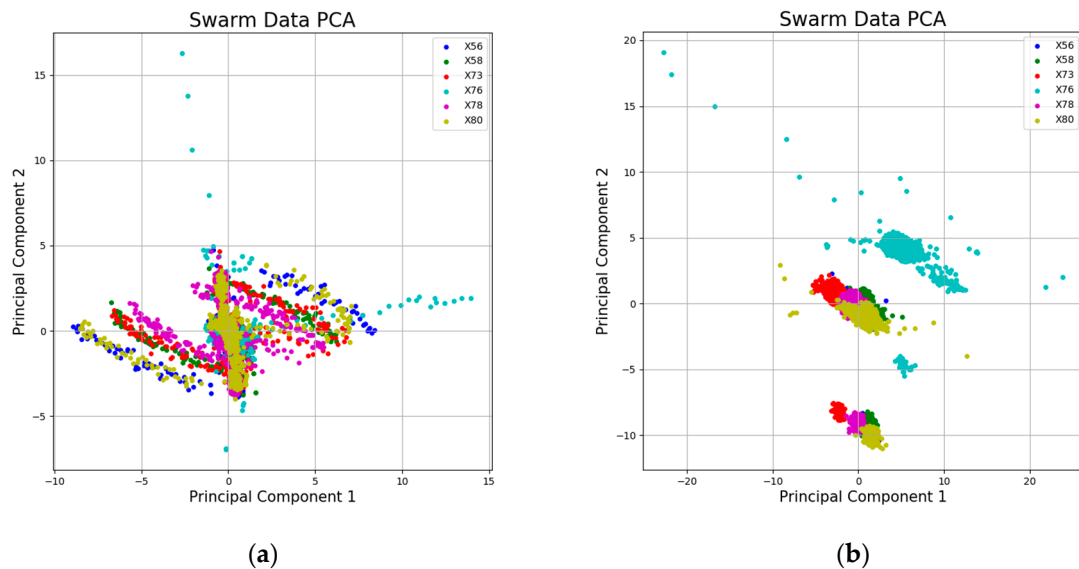
## 5. Numerical Results

This section presents illustrative case studies for the proposed methodology using the flight test data. Section 5.1 summarizes the results of PCA-based clustering; the generalization performance of the proposed classification method is reported in Section 5.2. Section 5.3 demonstrates the applicability of the proposed method as a means to monitor anomaly on the fly.

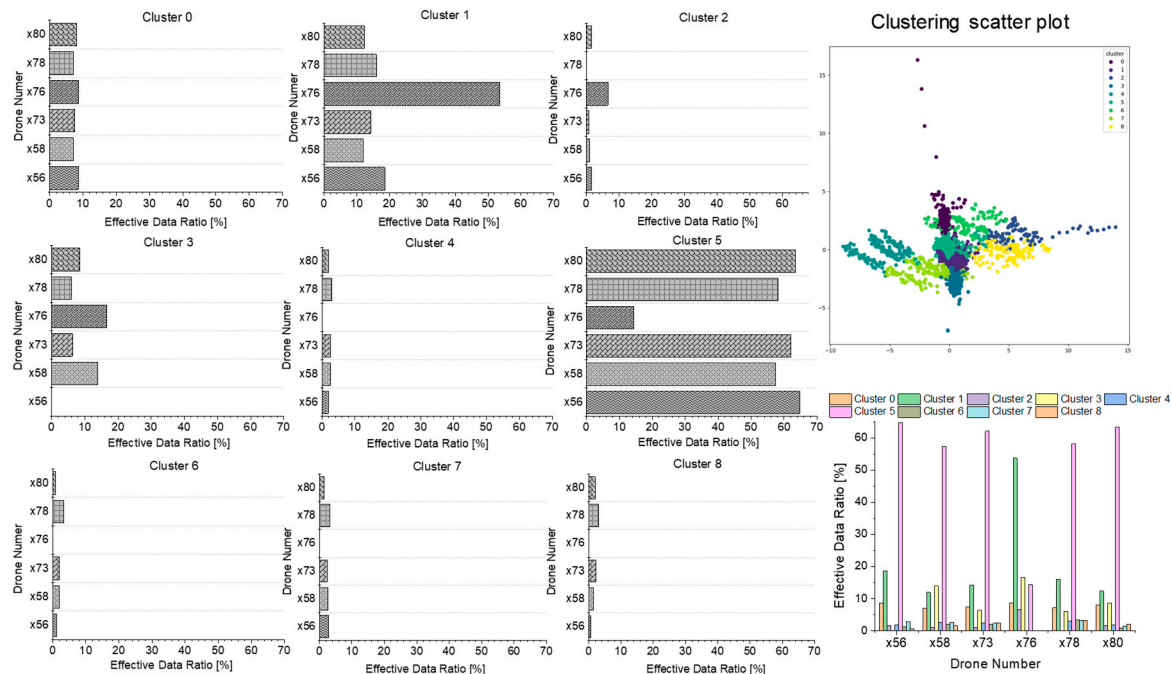
### 5.1. PCA and Clustering

Figure 4 represents PCA and K-means clustering results for data on a certain illustrative day, 25 May 2018. The distribution of the data points in the reduced space defined by the first and the second principal components are shown; the left plot is based on the RTK-GPS data and the right plot is based on the INS data. The dimension is reduced to principal component axes with a rate of cumulative dispersion of at least 90% through the PCA process, and the clusters are identified by K-means clustering. The two plots in Figure 5 depict the distribution of data points in the reduced space defined by the first and the second principal components. It can be seen in both plots that the data for one drone (x76) is distributed in a significantly different way from those for the other drones. Thus, it can be conjectured that data from x76 are likely to contain the time series that exhibits anomaly. However, it is not directly apparent how the clustering result is related to an anomaly, as it is not likely that all the data from a potentially faulty vehicle belong to a single cluster. Figure 5 represents what portion of data of each drone belong to a particular cluster. It can be seen that most of data from all the drones belong to Clusters 1 and 5, but the distributions between x76 and the others are different. According to the ratio of data belonging to Clusters 1 and 5, x76 behaves differently from the others—significantly more data belong to Cluster 1 than 5, as opposed to the other drones. Another noticeable observation is that any data from x76, potentially a problematic drone, do not belong to Clusters 4, 6, 7, and 8. As such, it can be conjectured that although the clustering does not clearly indicate which drone may have behaved abnormally, the distribution of data across the clusters may

be an indicator of anomaly in the data. Clusters 1, 2, and 3 are the indicators for abnormal data and Clusters 4, 5, 6, 7, and 8 indicate normal data. Along this line, data from each drone in Clusters 1~3 are labeled “TA” and the other data in Clusters 4~8 are labeled “TN” for the further procedure. In case of indistinct data such as Cluster 0 and uncertain data like as x56 in Cluster 3, a human expert may check the flight data to use as labeled data.



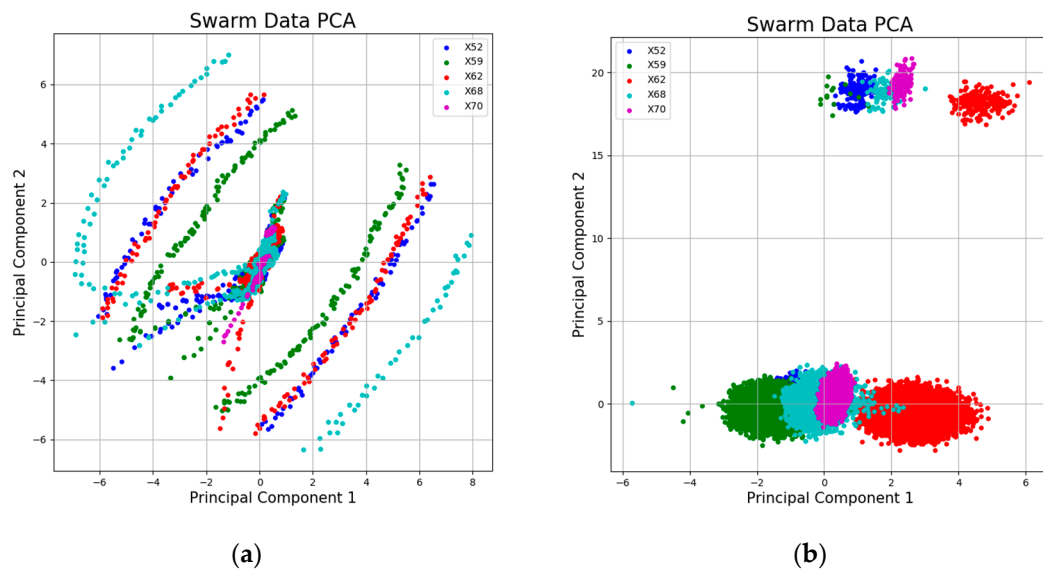
**Figure 4.** Principal component analysis (PCA) results data of Day 25 May 2018: (a) real-time kinematic global positioning system (RTK-GPS) and setpoint data; (b) sensor signals to detect anomalies in vehicle's inertial-navigation-system data. (The legend indicates particular drone IDs).



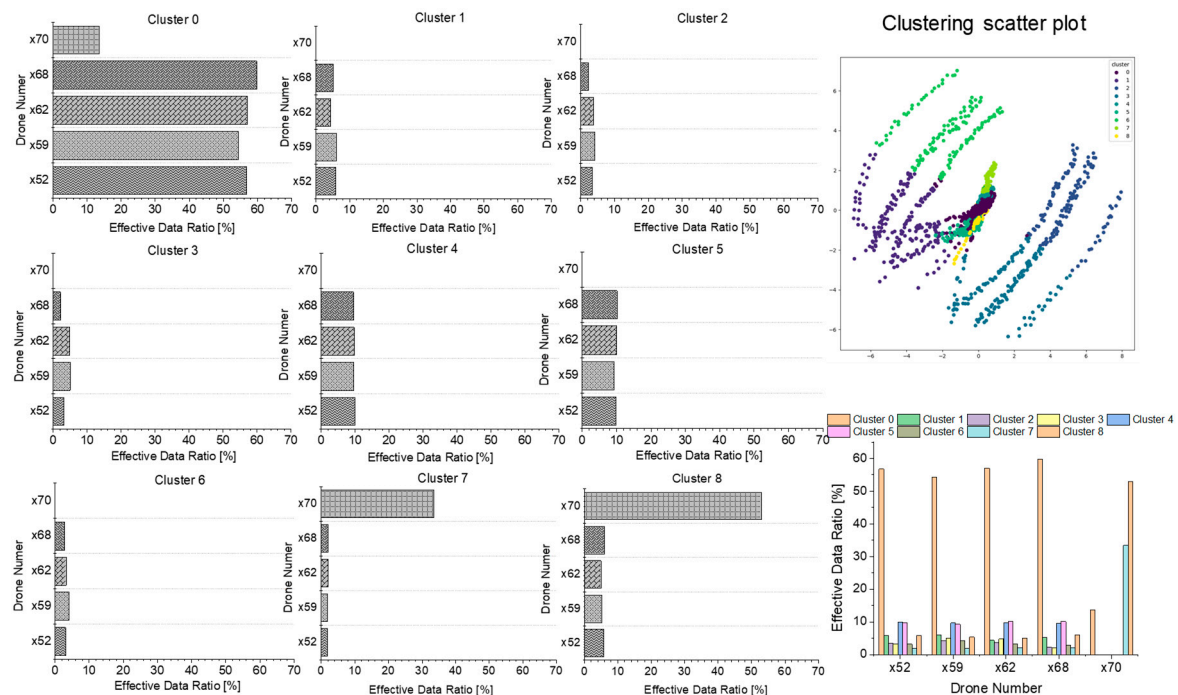
**Figure 5.** K-means clustering results for swarm-flight data (from 25 May 2018).

Figures 6 and 7 represent equivalent results for the data obtained from a flight test on another date, 5 June 2018. From the distribution of data point across the clusters, it can be first found that data from x70 exhibit very different characteristics from the others—the majority of this data belong to Clusters 0~6 to which only small portion of data from the other drones belong. On the other hand,

significantly more data for  $\times 70$  belong to Clusters 7 and 8. Thus, we can predict  $\times 70$  includes some anomaly behavior. From the clustering results, data from each drone in Clusters 0~6 are labeled “TA” and the others in Clusters 7 and 8 are labeled “TN”. While for the space reason this article only reports a couple of representative cases, a similar PCA and clustering procedure is done on all the flight tests for labeling. It should be noted that the clustering method herein does not aim for fully automated labeling but supporting an auxiliary means to aid human experts to conveniently label a large amount of unlabeled data.



**Figure 6.** PCA results data of Day 5 June 2018: (a) RTK-GPS and setpoint data; (b) Sensor signals to detect anomalies in vehicle's inertial-navigation-system data. (The legend indicates particular drone IDs).

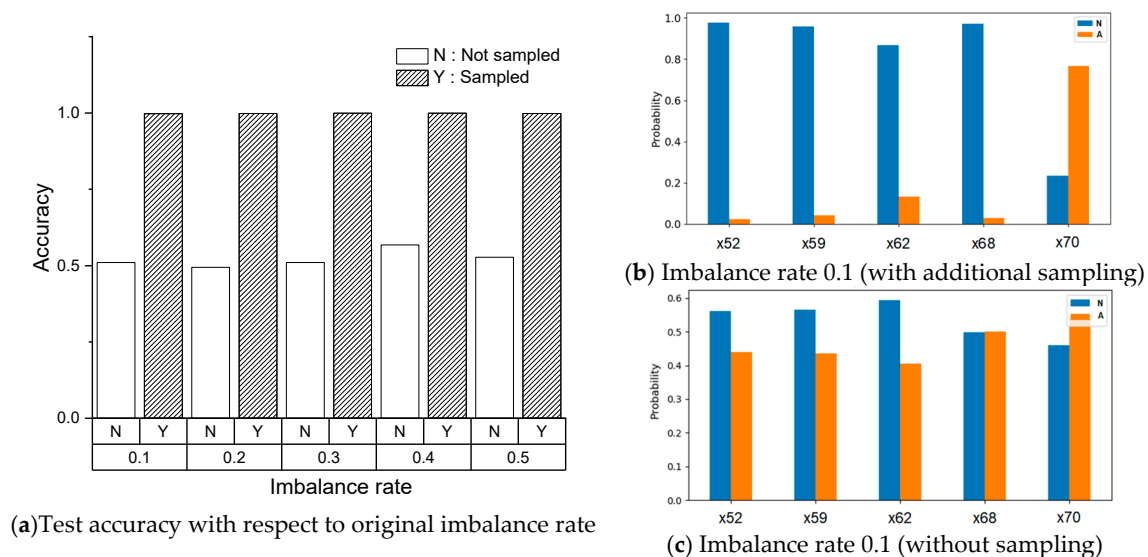


**Figure 7.** K-means clustering results for swarm-flight data (from 5 June 2018).

## 5.2. Classification Accuracy

The proposed 1D-CNN-based classifier is trained until the cross-entropy loss value converges with the learning parameters tabulated in Table 2. Flight test data obtained on several dates are used as the training set of size 65,989; 80% of this set is used for training the neural network and 20% for validation. Flight data obtained on the other two days are used as a test set of size 31,846.

To verify the relevance of the sampling scheme described in Section 4.4., the test accuracy with test data (5 June 2018) is compared with varying imbalance rate, i.e., the ratio of the abnormal data to the normal data in the original dataset. When the proposed sampling is not implemented, the abnormal data are used in the learning process with this imbalance rate compared to the normal data; if the sampling is applied, the abnormal data set is copied by the factor of the inverse of imbalance rate. Figure 8a compares the classification accuracy, defined as the number of correct classification over the total cases, depending on the use of sampling scheme for difference imbalance rate of the original data set. Figure 8b,c compares the anomaly detection results for the case when the original imbalance rate is 0.1. This is the case when the x70 drone exhibits an abnormal flight behavior; it can be seen that only the result with sampling clearly indicates anomaly in x70 data. It can be clearly seen that the proposed sampling technique significantly improves the classification accuracy, achieving zero classification error. It can be also noted that classification accuracy is around 50% when the sampling scheme is not implemented. One thing to note is that this is not improved even for larger imbalance rate of 0.5.

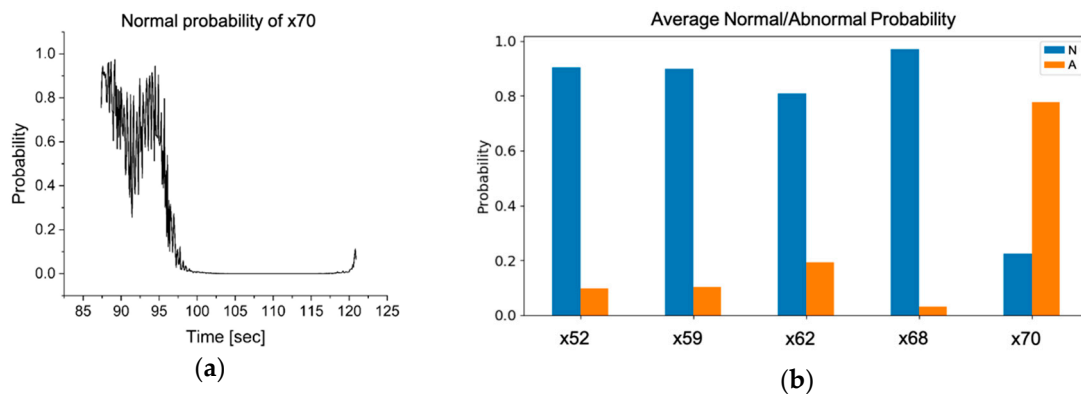


**Figure 8.** Effect of imbalance rate and sampling: (a) test accuracy with respect to original imbalance rate; (b,c) illustrative anomaly detection result for imbalance rate 0.1 (N: normal, A: abnormal).

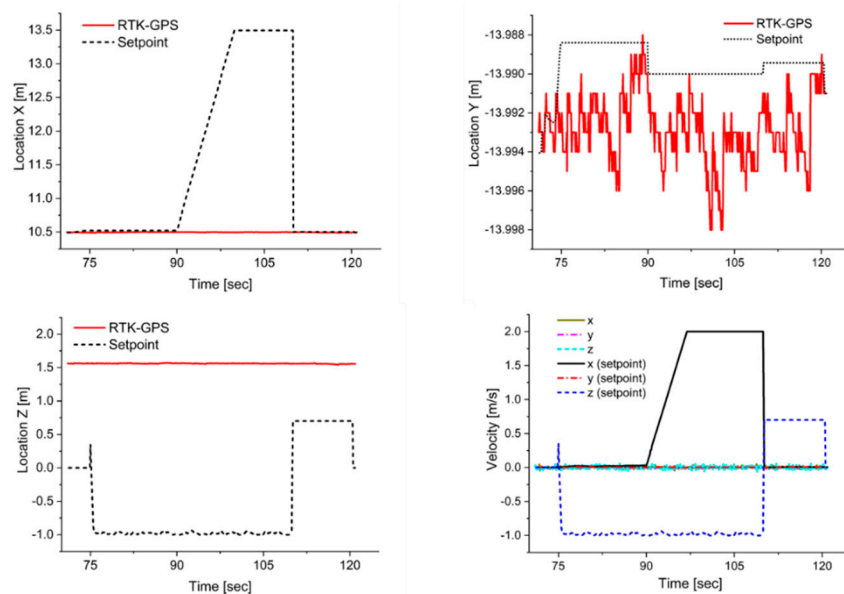
## 5.3. Anomaly Detection

The overall anomaly detection scheme is verified by checking whether or not the learned neural network classifier provides a valid indication for the time series input from the real flight test data. Figures 9 and 10 represent the results for a representative test data case when the input time series is not included in the training set. The swarm flight test data on 5 June 2018, whose clustering results are shown in Figures 6 and 7, are considered. Figure 9a depicts the output value of the neural network corresponding to the data of drone ID x70 that turns out to be in failure, when the input time series is continuously entered into the network. As the 1D-CNN takes an input of length 160 and the data rate is 10 Hz, the first output is obtained in response to the kinematic variables over the first time segment of 16 s. Then, the neural network continuously computes the output in response to the latest 160 time points. It can be found that the probability of being normal is oscillating for some period of time and then becomes zero constantly. Figure 9b compares the average value of the normal probability of the

data of five drones flown in this flight test, when the average is taken over the whole flight time. It can be clearly seen that the abnormal probability of this problematic drone is much greater than the normal probability. Figure 10 depicts the time history of the kinematic input variables corresponding to  $\times 70$ . Note that this drone does not effectively respond to the changes in the setpoints in  $x$ - and  $z$ -directions around the times 90 s and 75 s, respectively; as such, the anomaly detection scheme shown in Figure 9 indicates that the system may not be normal from the initial data point at time 85 s, which is obtained based on the behavior of the past 16 s; then, from time 100 s and onwards, it clearly says that the system is in the abnormal state with almost probability 1. It should be pointed out that investigation of the original flight log indicates that drone  $\times 70$  actually crashed during the flight test.



**Figure 9.** Anomaly classification result: (a) normal probability of  $\times 70$ , (b) average probability of vehicles.



**Figure 10.** Time history of kinematic variables of drone  $\times 70$ .

From the perspective of on-line anomaly monitoring, the two plots in Figure 9 may be considered as two extreme ways of monitoring. Continuous monitoring of the normal or abnormal probability as in Figure 9a provides the assessment of system anomaly in a high-frequency manner. This approach allows for responsiveness in the anomaly detection as it continuously provides updated information on the normality of the system. However, as can be seen in the figure, the output signal is particularly noisy in the transient phase. The average probability over the entire time span, as shown in Figure 9b provides a smoothed/collective decision on anomaly in the system. This approach may be least sensitive to noise, but the decision frequency may be too low.

To mitigate the noise in continuous monitoring at the same time to allow for responsive detection of anomaly, this work investigates moving average-based monitoring with finite time window. The normal/abnormal probability is averaged over some specified time horizon, and is updated with a new set of data. The time window of the adjacent averaging window can be overlapped. The four subplots of Figure 11 represent the change in the normal probability of different drones as the time proceeds. The average is taken over 30 s, and every 20 s the average value is updated, meaning that 1/3 of data are overlapped between the consecutive time windows. It can be observed that the abnormal probability of x70 is about 40% in the first time window, increases to about 55% in the second window, and eventually becomes greater than 90% in the third time window. This way, the anomaly can be monitored in a relatively high frequency without being subject to noise.

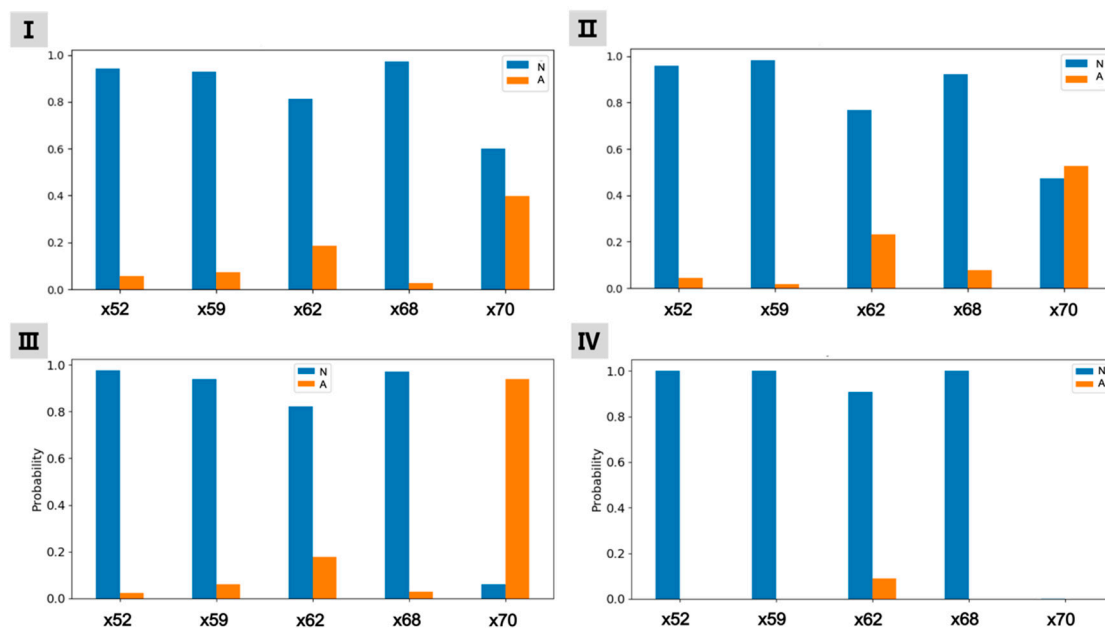


Figure 11. Moving average monitoring example.

## 6. Conclusions

This paper has presented a machine learning-based anomaly detection scheme for swarm drone flights. The proposed method features two major steps: the labeling step to label the unlabeled data based on lower-dimensional features, and the binary classification step based on a one-dimensional convolutional neural network with cross-entropy loss function. The deep neural network is trained and verified with real flight test data. It is also demonstrated that moving horizon-based monitoring can be a viable option for on-line monitoring of a system anomaly with a mitigated noise effect. Future work will include integration of anomaly monitoring in the overall health-management framework of swarm drones.

**Author Contributions:** Conceptualization, H.A. and H.-L.C.; Investigation, H.A. and S.T.M.; Methodology, H.A. and H.-L.C.; Data curation, S.M. and H.A.; Software, H.A. and M.K.; Visualization, H.A. and M.K.; Writing—original draft, H.A.; Writing—review and editing, H.A., and H.-L.C.; Supervision, H.-L.C.

**Funding:** This research was funded by Korea Aerospace Research Institute (grant number 1711094369, Development of Artificial Intelligence-Based Anomaly Detection Method for Aerospace Systems).

**Conflicts of Interest:** The authors declare no conflict of interest.



## Abbreviations

The following abbreviations are used in this manuscript:

AD	Anomaly detection
AI	Artificial intelligence
CNN	Convolutional neural network
GPS	Global positioning system
INS	Inertial navigation system
KNN	K nearest neighbor
LR	Logistical regression
ML	Machine learning
PCA	Principal component analysis
PHM	Prognostics and health monitoring
RNN	Recurrent neural network
RTK	Real time kinematic
TA	True anomaly
TN	True normal
UA	Uncertain anomaly
UN	True normal

## References

1. Otto, A.; Agatz, N.; Campbell, J.; Golden, B.; Pesch, E. Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. *Networks* **2018**, *72*, 411–458. [\[CrossRef\]](#)
2. Shakhathreh, H.; Sawalmeh, A.H.; Al-Fuqaha, A.; Dou, Z.; Almaita, E.; Khalil, I.; Othman, N.S.; Khreishah, A.; Guizani, M. Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges. *IEEE Access* **2019**, *7*, 48572–48634. [\[CrossRef\]](#)
3. Hu, J.; Lanzon, A. An innovative tri-rotor drone and associated distributed aerial drone swarm control. *Robot. Auton. Syst.* **2018**, *103*, 162–174. [\[CrossRef\]](#)
4. Cai, G.; Dias, J.; Seneviratne, L. A survey of small-scale unmanned aerial vehicles: Recent advances and future development trends. *Unmanned Syst.* **2014**, *2*, 175–199. [\[CrossRef\]](#)
5. Dydek, Z.T.; Annaswamy, A.M.; Lavretsky, E. Adaptive control of quadrotor UAVs: A design trade study with flight evaluations. *IEEE Trans. Control Syst. Technol.* **2012**, *21*, 1400–1406. [\[CrossRef\]](#)
6. Shrit, O.; Martin, S.; Alagha, K.; Pujolle, G. A new approach to realize drone swarm using ad-hoc network. In Proceedings of the 2017 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), Budva, Montenegro, 28–30 June 2017; pp. 1–5.
7. Bürkle, A.; Segor, F.; Kollmann, M. Towards autonomous micro uav swarms. *J. Intell. Robot. Syst.* **2011**, *61*, 339–353. [\[CrossRef\]](#)
8. Sahingoz, O.K. Networking models in flying ad-hoc networks (FANETs): Concepts and challenges. *J. Intell. Robot. Syst.* **2014**, *74*, 513–527. [\[CrossRef\]](#)
9. Shao, Z.; Yan, F.; Zhou, Z.; Zhu, X. Path planning for Multi-UAV formation rendezvous based on distributed cooperative particle swarm optimization. *Appl. Sci.* **2019**, *9*, 2621. [\[CrossRef\]](#)
10. Schilling, F.; Lecoeur, J.; Schiano, F.; Floreano, D. Learning vision-based cohesive flight in drone swarms. *arXiv* **2018**, arXiv:1809.00543.
11. Cho, D.-H.; Ha, J.-S.; Lee, S.; Moon, S.; Choi, H.-L. Informative path planning and mapping with multiple uavs in wind fields. In *Distributed Autonomous Robotic Systems*; Springer: Berlin, Germany, 2018; pp. 269–283.
12. Xie, S.; Zhang, A.; Bi, W.; Tang, Y. Multi-UAV mission allocation under constraint. *Appl. Sci.* **2019**, *9*, 2184. [\[CrossRef\]](#)
13. Yan, F.; Zhu, X.; Zhou, Z.; Chu, J. A hierarchical mission planning method for simultaneous arrival of multi-UAV coalition. *Appl. Sci.* **2019**, *9*, 1986. [\[CrossRef\]](#)
14. Bertuccelli, L.; Choi, H.-L.; Cho, P.; How, J. Real-time multi-UAV task assignment in dynamic and uncertain environments. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Kissimmee, FL, USA, 8–12 January 2018; p. 5776.

15. Choi, H.-L.; How, J. Coordinated targeting of mobile sensor networks for ensemble forecast improvement. *IEEE Sens. J.* **2011**, *11*, 621–633. [[CrossRef](#)]
16. Nagi, J.; Giusti, A.; Gambardella, L.M.; Di Caro, G.A. Human-swarm interaction using spatial gestures. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 september; pp. 3834–3841.
17. Hocraffer, A.; Nam, C.S. A meta-analysis of human-system interfaces in unmanned aerial vehicle (UAV) swarm management. *Appl. Ergon.* **2017**, *58*, 66–80. [[CrossRef](#)]
18. Hussein, A.; Abbass, H. Mixed initiative systems for human-swarm interaction: Opportunities and challenges. In Proceedings of the 2018 2nd Annual Systems Modelling Conference (SMC), Canberra, Australia, 4 October 2018; pp. 1–8.
19. Hussein, A.; Ghignone, L.; Nguyen, T.; Salimi, N.; Nguyen, H.; Wang, M.; Abbass, H.A. Towards bi-directional communication in human-swarm teaming: A survey. *arXiv* **2018**, arXiv:1803.03093.
20. Vistbakka, I.; Majd, A.; Troubitsyna, E. Multi-layered approach to safe navigation of swarms of drones. In Proceedings of the International Conference on Computer Safety, Reliability, and Security, Västerås, Sweden, 18–21 September 2018; pp. 112–125.
21. Qin, L.; He, X.; Zhou, D. A survey of fault diagnosis for swarm systems. *Syst. Sci. Control Eng. Open Access J.* **2014**, *2*, 13–23. [[CrossRef](#)]
22. Higgins, F.; Tomlinson, A.; Martin, K.M. Survey on security challenges for swarm robotics. In Proceedings of the 2009 Fifth International Conference on Autonomic and Autonomous Systems, Valencia, Spain, 20–25 April 2009; pp. 307–312.
23. Champion, M.; Ranganathan, P.; Faruque, S. A Review and Future Directions of UAV Swarm Communication Architectures. In Proceedings of the 2018 IEEE International Conference on Electro/Information Technology (EIT), Rochester, MI, USA, 3–5 May 2018; pp. 0903–0908.
24. Alfeo, A.L.; Cimino, M.G.; De Francesco, N.; Lega, M.; Vaglini, G. Design and simulation of the emergent behavior of small drones swarming for distributed target localization. *J. Comput. Sci.* **2018**, *29*, 19–33. [[CrossRef](#)]
25. Marzat, J.; Piet-Lahanier, H.; Damongeot, F.; Walter, E. Model-based fault diagnosis for aerospace systems: A survey. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2012**, *226*, 1329–1360. [[CrossRef](#)]
26. Boskovic, J.D.; Li, S.-M.; Mehra, R.K. On-line failure detection and identification (FDI) and adaptive reconfigurable control (ARC) in aerospace applications. In Proceedings of the 2001 American Control Conference (Cat. No. 01CH37148), Arlington, VA, USA, 25–27 June 2001; pp. 2625–2626.
27. Wang, Y.; Wang, D.; Wang, J. A data driven approach for detection and isolation of anomalies in a group of UAVs. *Chin. J. Aeronaut.* **2015**, *28*, 206–213. [[CrossRef](#)]
28. Wang, J.-H.; Wang, Y.-X.; Zhu, Y.-H. Bias compensation estimation in multi-UAV formation and anomaly detection. *J. Control Syst. Eng.* **2016**, *4*, 40–50. [[CrossRef](#)]
29. Freeman, P.; Pandita, R.; Srivastava, N.; Balas, G.J. Model-based and data-driven fault detection performance for a small UAV. *IEEE/ASME Trans. Mechatron.* **2013**, *18*, 1300–1309. [[CrossRef](#)]
30. Ordoukhanian, E.; Madni, A.M. Resilient multi-UAV operation: Key concepts and challenges. In Proceedings of the 54th AIAA Aerospace Sciences Meeting, San Diego, CA, USA, 4–8 January 2016; p. 0475.
31. Tagarev, T.; Sharkov, G.; Stoianov, N. Cyber security and resilience of modern societies: A research management architecture. *Inf. Secur.* **2017**, *38*, 93–108. [[CrossRef](#)]
32. Ordoukhanian, E.; Madni, A.M. System Trade-offs in Multi-UAV Networks. In Proceedings of the AIAA SPACE 2015 Conference and Exposition, Pasadena, CA, USA, 31 August–2 September 2015; p. 4542.
33. Negash, L.; Choi, H.-L. Distributed observers for cyberattack detection and isolation in formation flying UAVs. *J. Aerosp. Inf. Syst.* **2017**, *14*, 551–565.
34. Suarez, A.; Heredia, G.; Ollero, A. Cooperative sensor fault recovery in multi-UAV systems. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1188–1193.
35. Li, G.; Rai, A.; Lee, H.; Chattopadhyay, A. Operational anomaly detection in flight data using a multivariate gaussian mixture model. In Proceedings of the Annual Conference of the PHM Society, Philadelphia, PA, USA, 24–27 September 2018.

36. Nanduri, A.; Sherry, L. Anomaly detection in aircraft data using Recurrent Neural Networks (RNN). In Proceedings of the 2016 Integrated Communications Navigation and Surveillance (ICNS), Herndon, VA, USA, 19–21 April 2016; pp. 5C2-1–5C2-8.
37. Manukyan, A.; Olivares-Mendez, M.A.; Voos, H.; Geist, M. Real time degradation identification of UAV using machine learning techniques. In Proceedings of the 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 13–16 June 2017; pp. 1223–1230.
38. Yong, D.; Yuanpeng, Z.; Yaqing, X.; Yu, P.; Datong, L. Unmanned aerial vehicle sensor data anomaly detection using kernel principle component analysis. In Proceedings of the 2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI), Yangzhou, China, 20–22 October 2017; pp. 241–246.
39. Jolliffe, I.T.; Cadima, J. Principal component analysis: A review and recent developments. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2016**, *374*, 20150202. [\[CrossRef\]](#)
40. Kiranyaz, S.; Avci, O.; Abdeljaber, O.; Ince, T.; Gabbouj, M.; Inman, D.J. 1D convolutional neural networks and applications: A survey. *arXiv* **2019**, arXiv:1905.03554.
41. Hendrycks, D.; Mazeika, M.; Kadavath, S.; Song, D. Using self-supervised learning can improve model robustness and uncertainty. In Proceedings of the Thirty-third Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8 December–14 December 2019.
42. Moon, S.; Choi, Y.; Kim, D.; Seung, M.; Gong, H. Outdoor Swarm Flight System Based on RTK-GPS. *J. Kiise* **2016**, *43*, 1315–1324. [\[CrossRef\]](#)
43. Brillinger, D.R. *Time Series: Data Analysis and Theory*; Siam: Philadelphia, PA, USA, 1981; Volume 36.
44. Selim, S.Z.; Ismail, M.A. K-means-type algorithms: A generalized convergence theorem and characterization of local optimality. *IEEE Trans. Pattern Anal. Mach. Intell.* **1984**, *PAMI-6*, 81–87. [\[CrossRef\]](#)
45. Chollet, F. *Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek*; MITP-Verlags GmbH & Co. KG: Frechen, Germany, 2018.
46. Dreiseitl, S.; Ohno-Machado, L. Logistic regression and artificial neural network classification models: A methodology review. *J. Biomed. Inform.* **2002**, *35*, 352–359. [\[CrossRef\]](#)
47. Kotsiantis, S.; Kanellopoulos, D.; Pintelas, P. Data preprocessing for supervised learning. *Int. J. Comput. Sci.* **2006**, *1*, 111–117.
48. Famili, A.; Shen, W.-M.; Weber, R.; Simoudis, E. Data preprocessing and intelligent data analysis. *Intell. Data Anal.* **1997**, *1*, 3–23. [\[CrossRef\]](#)
49. Bholowalia, P.; Kumar, A. EBK-means: A Clustering Technique Based on Elbow Method and K-Means in WSN. *Int. J. Comput. Appl.* **2014**, *105*. Available online: <https://www.ijcaonline.org/archives/volume105/number9/18405-9674> (accessed on 12 December 2019).
50. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep neural network training by reducing internal covariance shift. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015; Volume 37, pp. 448–456.
51. Japkowicz, N.; Stephen, S. The class imbalance problem: A systematic study. *Intell. Data Anal.* **2002**, *6*, 429–449. [\[CrossRef\]](#)
52. Kingma, D.P.; Ba, J.L. Adam: A method for stochastic optimization. *arXiv* **2017**, arXiv:1412.6980v9.
53. Vinod, N.; Geoffrey, E.H. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML), Haifa, Israel, 21–24 June 2010; pp. 807–814.
54. Lin, M.; Chen, Q.; Yan, S. Network in network. *arXiv* **2013**, arXiv:1312.4400.

