International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 20 (2019) pp. 3946-3952

Using the Statistical Features of the Data to Detect Potential Failure of Unmanned Aerial Vehicles

Ahmad M. Alos[1], Zouhair. Dahrouj[2]

[1]*Higher Institute for Applied Sciences and Technology/ Informatics Department, Damascus, Syria.*

[2]*Higher Institute for Applied Sciences and Technology/ Informatics Department, Damascus, Syria.*

Abstract

The Unmanned Aerial Vehicle is one of the most complex systems ever developed. Its complexity raises the chances of its failure. This paper focuses on predicting the potential failure of the UAV using the collected data from its previous missions. The collected data consist of discrete and continuous attributes. The proposed approach helps in determining the abnormal flights, and the contribution of the attributes in the potential faults. The values of the attributes are analyzed using some statistical parameters. The selected parameters are {Mean, Variance, Standard deviation, Kurtosis, Skewness, Minimum, and Maximum}. These parameters are used to build feature datasets to characterize the performed flights. Next, two algorithms can be used to extract the anomalies from the feature datasets. The used algorithms are the Principal Components Analysis-based anomaly detector and the One-Cluster K-Means. The conducted experiments showed that our approach is practical for detecting the faults and the contributed attributes using either discrete or continuous data.

Keywords: UAV, Statistical, Anomaly detection, K-Means, Principal Components Analysis.

I. INTRODUCTION

Anomaly detection applications include intrusion, fraud detection, medical applications, and robot behavior. These live applications motivated many researchers to work in this area over recent years. Unmanned Aerial Vehicle (UAV) is an aircraft piloted by remote control or onboard computers. Its applications have been increasing in recent years, including surveillance, reconnaissance, aerial photography, and disaster monitoring [1]. It is a very complex system operated by Control, Aerodynamics, Communications, and Informatics. The complexity of the UAV system raises the chances of its failure. Anomaly detection algorithms predict system failure by finding patterns in data that do not conform to expected behavior [2]. These algorithms operate in three modes [2][3]: Supervised, semi-supervised, and unsupervised. The supervised anomaly detection algorithms assume the availability of training data with given labels for the normal class as well as for the anomalous class. The semi-supervised anomaly detectors assume the training data has labeled instances for the normal class only. The unsupervised anomaly detectors do not require any labeled training data. They implicitly assume that normal instances are far more frequent than anomalies in the test data.

The history of the UAV flight missions contains the values of many input attributes (Commands), and sensor readings (UAV state). The commands could be an Altitude command, Rudder Command, Aileron Command, Throttle..., and the sensor readings could be pitch, roll, yaw, longitude, latitude, altitude, and so on. The values of these attributes are statistically analyzed using a wide range of statistical parameters. These parameters behave differently in case of a potential fault, and the combined anomalous values of the statistical parameters help to predict that potential fault. Our contribution is a new technique to detect irregularities in the UAV behavior by monitoring these statistical features in order to assess the chances of system failure based on data from previous missions. The proposed technique builds a feature dataset for each attribute. Each element in each feature dataset is related to a flight mission. Next, the technique detects the behavior irregularities in each flight by using one of two famous anomaly detectors: (1) The PCA (Principal Component Analysis) anomaly detector, and (2) The One-Cluster K-Means classifier. The goal of the proposed approach is to detect the abnormal flight missions, predict potential failure, and the contribution of the different attributes in the fault.

The rest of the paper is organized as follows. Section II provides a brief background and a review of related works. Problem description and the features used in our approach are explained in section III. Also, section III explains PCA-based anomaly detection and the One-Cluster K-Means algorithms. The used dataset, the results of the experiments, and a comparison with the MKAD method (Multiple Kernel based Anomaly Detection) are explained in section IV. Finally, section V describes the conclusion and future suggestions.

II. RELATED WORK

System faults can be predicted by detecting anomalies in the system data. There are three types of fault detection algorithms [4]: Model-based, Knowledge-based, and Data-driven-based algorithms. Cork et al. [5] used a model-based fault detection algorithm, where they estimated the state of the UAV using a nonlinear dynamic model. They used the divergence of the estimated state from its actual value to detect system faults. The knowledge-based algorithms depend on predefined rules (if-then) sentences. Bu et al. [6] developed a method to detect the faults of the UAV sensors using a fuzzy logic model. The data-driven algorithms depend on the statistical information to detect outliers and label them as faults. Lin et al. [7] designed an online algorithm to detect UAV sensor faults based on a statistical analysis of sensor readings and navigation data. Sun et al. [8] were interested in data-driven algorithms because model-based and knowledge-based algorithms had rule dependencies. Knowledge-based algorithms are unable to detect unknown or non-modeled faults, while data-driven algorithms showed flexibility due to the model-free analysis. Lin. [7], Khalastchi. [9], and Pokrajac [10] used statistical methods that produced an anomaly score for each given point of time. Each anomaly score depends on a sliding window of monitored readings. Most of the methods considered the point density using either Mahalanobis Distance, K-Nearest Neighbor (KNN), or K-Means classifier. PCA (Principal Components Analysis) is a fast technique used for dimensionality reduction. It reduces the dimensionality of a multivariate data set into two or three attributes. Paffenroth et al. [12] developed a PCA-based anomaly detector to predict cyber-network attacks. Yong et al. [13] used PCA to detect anomalies in large sample size and complicated relationships of sensor data. S. Das et al. [14] developed the MKAD (Multiple Kernel Anomaly Detection) algorithm to ensure the safety of an aviation system. The MKAD method is builds different multiple kernel functions for both discrete and continuous values. Kernel functions map pairs of objects to their similarity. The values of the similarity range between One and Zero, where value one is given for maximum similarity and value zero is given for no similarity. The MKAD method uses the One-class Support Vector Machine (SVM) to separate the abnormal flights from the normal ones. Building the SVM model required a large training dataset, which is considered time-consuming and memory demanding [15].

III. METHODS

A. Definitions

$F_n: n \in \{1, 2, \dots, N\}$ denotes a flight mission, where N is the total number of the available UAV flights. Each flight consists of multiple attributes. The values of these attributes are recorded at each step of time (t). Suppose that T denotes the end period of a flight, It is assumed that all the attribute do not have empty values [7]. Let $I = \{A_j: 0 \leq j < |I|\}$ denotes the set of all attributes. In a mission flight F_n, the values of the attribute $A_j \in I$ are stored in the vector $V_{n,j} = \{v_{j,0}, \dots, v_{j,t}, \dots, v_{j,T}\}$.

To analyze the values of each UAV attribute A_j statistically, we select some statistical parameters. The selected parameters are {Mean, Variance, Standard deviation, Kurtosis, Skewness, Minimum, and Maximum}. The standard deviation is the measurement of the amount of variability for a set of data from its mean. The variance is the squared value of the standard deviation. Kurtosis indicates the flatness or the spikiness of the dataset. Skewness characterizes the degree of the dataset asymmetry around its mean. The Minimum refers to the minimum value in a given set, and finally, the maximum refers to the maximum value in a given dataset [16]. In each flight F_n and for each attribute $A_j \in I$, formula (1) calculates the Mean, formula (2) calculates the Variance, formula (3) calculates the Standard Deviation, formula (4) calculates the Kurtosis, and formula (5) calculates the skewness.

$$Mean = \overline{v}_j = \frac{\sum_{t=0}^{T-1} v_{j,t}}{T}, \quad (1)$$

$$Variance = \frac{\sum_{t=0}^{T-1} (v_{j,t} - \overline{v}_j)^2}{T-1}, \quad (2)$$

$$Standard\ Deviation = s = \sqrt{\frac{\sum_{t=0}^{T-1} (v_{j,t} - \overline{v}_j)^2}{T-1}}, \quad (3)$$

$$Kurtosis = \langle \frac{T(T+1)}{(T-1)(T-2)(T-3)} \sum_{t=0}^{T^{n}-1} \frac{(v_{j,t} - \overline{v}_j)^4}{s_n} \rangle - \frac{3(T-1)^2}{(T-2)(T-3)}, \quad (4)$$

$$Skewness = \frac{T}{(T-1)(T-2)} \sum_{t=0}^{T-1} \frac{(v_{j,t} - \overline{v}_j)^3}{s} \quad (5)$$

B. Construct Feature Dataset

The proposed approach starts by building the feature datasets using the values of the selected statistical parameters. Algorithm 1 constructs the features datasets.

Algorithm 1 Construct_Feature_Datasets($A_j \in I$)

$R_j \leftarrow \emptyset$
for each F_n do
  $r_{n,j} \leftarrow$calculate[Mean, Variance, Kurtosis…] for A_j
  add $r_{n,j}$ to R_j
return R_j

Algorithm 1 calculates the values of the statistical parameters using formulas (1)-(5), then it adds these values as a row $r_{n,j}$ to the feature dataset R_j. The resulted feature dataset R_j is of size N rows, the same as the total number of the flights. To classify the instances of each R_j we use two different methods: (1) Principal Component Analysis (PCA)-based anomaly detector, and (2) The One-Cluster K-Means classifier.

A. PCA-based Anomaly Detection

The PCA-based anomaly detector is composed of two steps: Dimensionality Reduction [17] and Threshold Classification (see Algorithm 2). Principal Component Analysis projects a given set of data points onto a set of uncorrelated variables. These variables are called *Principal Components* (PC), which are ordered by the amount of data variance in descending order [18], and they are linear combinations of the original variables [19]. Applying PCA to the normalized data matrix Y in $\mathbb{R}^n$generates $\{v_i\}_{i=1}^{p}$ which is a set of p principal components. The first PC v_1 is the vector that corresponds to the direction of maximum variance [20], and it is denoted by formula (6).

$$v_1 = arg\ max_{\|x\|=1} \|Yx\| \tag{6}$$

$\|x\|$ is the 2-norm of x, and $\|Yx\|$ is proportional to the variance of the data distributed along x. The second PC is the linear combination of the original variables with the second-largest variance and orthogonal to the first PC [17]. Proceeding iteratively, if the previous $i-1$ principal components have been selected, the residual is the difference between the original samples and the samples corresponding to these $i-1$ PCs. Therefore, the i_{th} PC is defined as[20]

$$v_i = arg\ max_{\|x\|=1} \left\|(Y - \textstyle\sum_{j=1}^{i-1} Y v_j v_j^T)x\right\|. \tag{7}$$

The data matrix Y is of order $(m \times n)$, m is its length, and n is the count of the original variables. v_i denotes the i_{th} eigenvector of the estimated covariance matrix

$$A = \frac{1}{m} Y^T Y\ . \tag{8}$$

PCA in the feature space calculates the eigenvalues and eigenvectors according to the following equation [13]:

$$\lambda v_i = A v_i, \tag{9}$$

and the projection of the data onto each principal component is given by

$$u_i = v_i^T, i = 1,2, \dots, p, \tag{10}$$

The projections of instance x on the principal components are $u1, u2, \dots, u_n$ where the corresponding Eigen-values are $\lambda_1, \lambda_2, \dots, \lambda_n$. If u_{i+1} passes a defined threshold, then the first i principal components are regarded as normal; otherwise it is considered abnormal [2] [18]. We use PCA to reduce the dimensionality of each feature dataset R_j . The result of dimensionality reduction is a new dataset D_j of N scores and one dimension. Choosing one dimension for the results is necessary to apply one threshold to classify the new instances in D_j into either normal or abnormal. Considering that for each dataset D_j the scores have different scales, we can apply the min-max normalization to normalize the scores using formula (11).

$$score_{normalized} = \frac{score_i - score_{min}}{score_{max} - score_{max}}. \tag{11}$$

The resulted normalized scores are stored in ND_j(see Algorithm 2). The values of ND_j range between zero, and One. Defining a suitable threshold for each dataset ND_j is achieved with the help of domain expertise, visualization, and the previous knowledge of the percent of the abnormal flights. The scores that exceed the defined threshold are considered abnormal flights. Algorithm 2 summarizes the PCA-based anomaly detection for each generated features dataset R_j

Algorithm 2 PCA_Based_Anomaly_Detector(R_j)
D_j ←Reduce the dimensionality of R_j
ND_j ←Normalize(D_j)
abnormal flights ←ThresholdClassification(ND_j)
return (abnormal flights)

B. K-Means Anomaly Detection

The K-Means algorithm is a clustering algorithm, which classifies a given dataset into K classes, where K is a predefined number. K-means learns the centroids of the clusters from the dataset, and it tries to minimize the following objective function[21]:

$$J_c = \textstyle\sum_{k=1}^{K} \sum_{X_n \in C_k} d(X_n, Z_k), \tag{12}$$

where K is the number of the clusters, X_n denotes the n_{th} element of the dataset. C_k is the k_{th} cluster, Z_k is its centroid, and $d(X_n, Z_k)$ is the distance between the element X_n and the centroid of the cluster Z_k. The widely used distance is the Euclidean distance. To use K-Means in anomaly detection, we set K to be one class (Normal class) [21]. The elements that do not belong to the one cluster C_1 are considered abnormals. Algorithm 3 shows the One-Cluster K-Means algorithm, where

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 20 (2019) pp. 3946-3952
© Research India Publications. http://www.ripublication.com

it tries to find the best cluster to classify the normal instances. The algorithm starts by selecting one element randomly to be the centroid, and then it adds the nearest element to the cluster C_1. Next, it recalculates the new centroid of the cluster using formula (13).

$$z_1 = \frac{1}{p}\sum_{r_m \in C_1} r_m, \tag{13}$$

r_m denotes the m_{th} element in the cluster C_1, and p is the number of elements in the cluster. The algorithm iterates until the centroids do not change anymore, or a maximum number of iterations have been reached. The abnormal flights are labeled by the elements that belong to the difference R_j/C_1 (i.e., the instances belong to R_j, and do not belong to the cluster C_1).

Algorithm 3 K-Means_Anomaly_Detector(R_j)

z_1 ←select randomly one point to be the centroid
repeat until the maximum iteration
  $C_1 \leftarrow \emptyset$.
  for each $r_{\mathrm{n,j}} \in R_j$
    if $d(r_{\mathrm{n,j}}, z_1)$< threshold then
      add $r_{\mathrm{n,j}}$ to C_1
  m_1 ←recalculate_the_centroid(C_1)
  if($z_1 = m_1$)
    stop
  $z_1 \leftarrow m_1$
abnormal ← R_j/C_1
return (abnormal)

IV. EXPERIMENTAL RESULTS

To test the algorithms, we used the publicly published MKAD synthetic dataset [14], which includes various types of seeded faults. This dataset was used to test the MKAD (Multiple Kernel based Anomaly Detection) method robustness [14]. The MKAD method is a state of the art algorithm that builds different multiple kernel functions for both discrete and continuous values. It uses the One-class Support Vector Machine (SVM) to separate the abnormal flights from the normal ones. The One-Class SVM (Support Vector Machine) learns a region (a boundary) that contains the training data instances. For each test instance, it determines if it belongs to the learned region. If the test instance falls within the learned region, then it is flagged as normal; otherwise, it is flagged as abnormal [2]. The One-Class SVM model is learned from a training dataset (150 training flights). The testing part of the MKAD synthetic dataset also includes 150 testing flights. The flights are labeled sequentially as {Flight00201, Flight00202…, Flight00350). Each flight includes 1000 rows and consists of 15 attributes. We labeled these attributes sequentially (A1, A2…, A15). The first ten attributes are of discrete values, and the last five are of continuous values. The MKAD dataset is injected with four types of faults (see Table I), where three examples of each fault were injected in the flights.

TABLE I. THE INJECTED FAULTS IN THE MKAD SYNTHETIC DATASET

Fault Type	Description
Fault type I	Missing expected values in discrete data (see Fig.1).
Fault type II	Extra-unexpected values in discrete data (see Fig.2).
Fault type III	Out of order sequences of values in the discrete data (see Fig.3).
Fault type IV	Abnormal patterns in continuous data (see Fig.4).

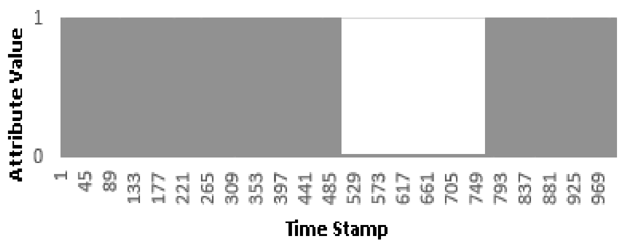

Figure 1. Fault type I (missing expected values).

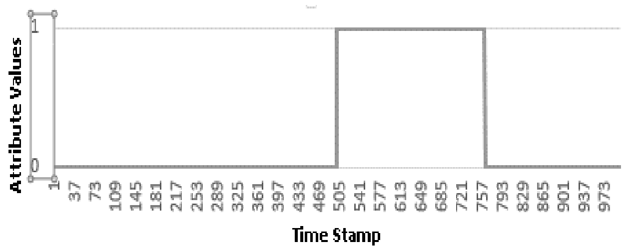

Figure 2. Fault type II (extra-unexpected values).

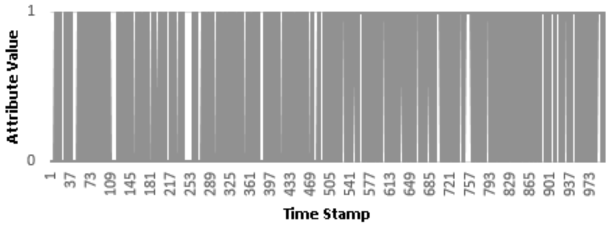

Figure 3. Fault type III: (out of order sequences of values).

.

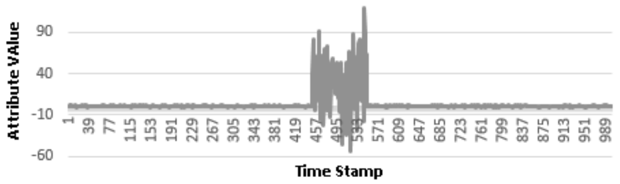

Figure 4. Fault type IV (abnormal patterns).

The first step is to extract the feature datasets. The implementation of Algorithm 1 generated 15 feature datasets R_j, where every dataset included the values of the selected statistical parameters [Mean, Variance, Kurtosis…] for each attribute A_j and through all the 150 flights. The next step is to classify the instances of the feature datasets using the proposed algorithms. Applying PCA dimensionality reduction produced 15 score datasets. For each score dataset, the PCA-based anomaly detector identified several abnormal flights. The threshold for each score dataset was defined visually and by using the previous knowledge of the percent of the abnormal flights, which is 8% of the total number of flights. This percent value can be extracted from the meta-data of the MKAD synthetic dataset.

The K-Means algorithm used the feature datasets directly and produced 15 other score datasets. K-Means algorithm did not need a threshold to classify the abnormal flights, while PCA needed a threshold for every score dataset. The PCA-based anomaly detector and the One-cluster K-Means algorithms detected 100% of the abnormal flights, with no false alarms (whether the fault was discrete or continuous). The two algorithms found the contributed attributes for each abnormal flight, except in flight "Flight00269", where K-Means found that the attribute A14 contributed in the detected fault, while PCA did not detect the contribution of attribute A14, because of the unsuitable threshold (0.4) (see Fig.5, Fig.6, and Table II). Our implementation of the two algorithms had similar processing time, although the PCA-based anomaly detector was a bit faster (see table III). The two algorithms had similar results with the MKAD method (The reader is referred to [14] to see the results of the MKAD method). However, the PCA-based anomaly detector and the K-Means algorithm did not need the training dataset since they are unsupervised algorithms, while the MKAD method required a large training dataset for building the kernel functions and the SVM model, which is considered time-consuming and memory demanding [15].

TABLE II. ABNORMAL FLIGHTS AND PERCENT OF CONTRIBUTED VARIABLES

Fault Type	**Flights**	**Contributed Attributes Found by PCA**	**Contributed Attributes Found by K-Means**
I	Flight00230	A7	A7
	Flight00298	A7	A7
	Flight00314	A7	A7
II	Flight00237	A1, A4, A7	A1, A4, A7
	Flight00260	A5, A7	A5, A7
	Flight00336	A1, A4, A7	A1, A4, A7
III	Flight00214	A2, A3, A6, A8, A9, A10	A2, A3, A6, A8, A9, A10
	Flight00238	A2,A3, A6, A8, A9, A10	A2,A3, A6, A8, A9, A10
	Flight00325	A2, A3, A6, A8, A9, A10	A2, A3, A6, A8, A9, A10
IV	Flight00233	A11, A12, A13, A14, A15	A11, A12, A13, A14, A15
	Flight00269	A11, A12, A13, A15	A11, A12, A13, **A14**, A15
	Flight00295	A11, A12, A13, A14, A15	A11, A12, A13, A14, A15

TABLE III. A COMPARISON BETWEEN PCA-BASED ANOMALY DETECTOR AND THE ONE-CLUSTER K-MEANS ALGORITHM

Algorithm	**Discrete Attributes Faults**	**Continuous Attributes Faults**	**Processing Time (milliseconds)**
PCA	100%	100%	1544
K-Means	100%	100%	1560

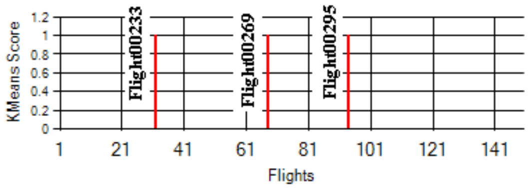

Figure 5. The One-Cluster K-Means score dataset for Attribute A14.

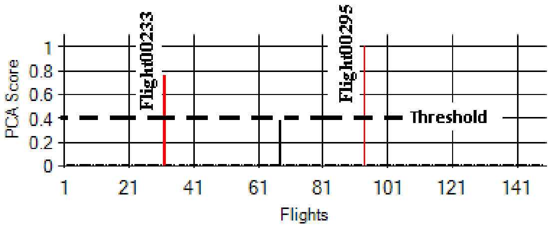

Figure 6. The PCA-based anomaly detector score dataset for Attribute A14.

V. CONCLUSION

We proposed a new approach to assess UAV flights as a method to ensure its safety. The approach extracts the values of some well-known statistical parameters. The used features are Mean, Variance, Standard deviation, Kurtosis, Skewness, Minimum, and Maximum. The parameters are collected for each attribute. The result is a set of feature datasets, where the rows of each dataset are related to a single flight. Then two algorithms (PCA-based anomaly detector and One-cluster K-Means classifier) are used to extract the anomaly instances. Each anomaly instance is used to label the related flight as an abnormal flight. At the same time, the two algorithms find the contributed attributes in the potential faults. The two proposed algorithms succeeded in detecting the faults, the abnormal flights, and the contributed attributes similarly. Besides, they showed similar results as the MKAD method (Multiple Kernel based Anomaly Detection). However, unlike the MKAD method, they did not need a considerable training set. Future enhancements could be (1) a better method for choosing the threshold of the PCA-based detector. (2) Other algorithms can be used for classification, such as neural networks, logistic regression…, or a combination of more than one algorithm.

ACKNOWLEDGMENT

This work is supported by a grant from the Higher Institute for Applied Sciences and Technology, Damascus, Syria. The authors appreciate the valuable comments provided by the anonymous referees.

REFERENCES

[1] I. Colomina and P. Molina, "Unmanned aerial systems for photogrammetry and remote sensing: A review," *ISPRS J. Photogramm. Remote Sens.*, vol. 92, pp. 79–97, 2014.

[2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, 2009.

[3] I. Renckens, "Artificial Intelligence Automatic Detection Of Suspicious Behaviour," p. 59, 2014.

[4] E. Khalastchi, M. Kalech, G. A. Kaminka, and R. Lin, "Online data-driven anomaly detection in autonomous robots," *Knowl. Inf. Syst.*, vol. 43, no. 3, pp. 657–688, 2015.

[5] L. Cork and R. Walker, "Sensor Fault Detection for UAVs using a Nonlinear Dynamic Model and the IMM-UKF Algorithm," in *Information, Decision and Control*, 2007, pp. 230–235.

[6] J. Bu *et al.*, "Integrated method for the UAV navigation sensor anomaly detection," *IET Radar, Sonar Navig.*, vol. 11, no. 5, pp. 847–853, May 2017.

[7] R. Lin, E. Khalastchi, and G. A. Kaminka, "Detecting anomalies in unmanned vehicles using the Mahalanobis distance," *Proc. - IEEE Int. Conf. Robot. Autom.*, no. June 2014, pp. 3038–3044, 2010.

[8] R. Sun, Q. Cheng, G. Wang, and W. Y. Ochieng, "A novel online data-driven algorithm for detecting UAV navigation sensor faults," *Sensors (Switzerland)*, vol. 17, no. 10, 2017.

[9] E. Khalastchi, G. a. Kaminka, M. Kalech, and R. Lin, "Online anomaly detection in unmanned vehicles," *10th Int. Conf. Auton. Agents Multiagent Syst. 1*, no. Aamas, pp. 115–122, 2011.

[10] D. Pokrajac, L. J. Latecki, and A. Lazarevic, "Incremental local outlier detection for data streams," *IEEE Symp. Comput. Intell. Data Min.*, no. April, pp. 504–515, 2007.

[11] A. Karami and M. Guerrero-Zapata, "A fuzzy anomaly detection system based on hybrid PSO-Kmeans algorithm in content-centric networks," *Neurocomputing*, vol. 149, no. PC, pp. 1253–1269, Feb. 2015.

[12] R. Paffenroth, K. Kay, and L. Servi, "Robust PCA for Anomaly Detection in Cyber Networks," Jan. 2018.

[13] D. Yong, X. Yaqing, Z. Yuanpeng, P. Yu, and L. Datong, "Unmanned aerial vehicle sensor data anomaly detection using kernel principal component analysis," in *ICEMI 2017 - Proceedings of IEEE 13th International Conference on Electronic Measurement and Instruments*, 2017, vol. 2018-Janua, pp. 241–246.

[14] S. Das, B. L. Matthews, A. N. Srivastava, and N. C. Oza, "Multiple kernel learning for heterogeneous anomaly detection," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '10*, 2010, p. 47.

[15] V. M. Janakiraman and D. Nielsen, "Anomaly detection in aviation data using extreme learning machines," *Proc. Int. Jt. Conf. Neural Networks*, vol. 2016-Octob, pp. 1993–2000, 2016.

[16] V. Sugumaran and K. I. Ramachandran, “Automatic rule learning using decision tree for fuzzy classifier in fault diagnosis of roller bearing,” *Mech. Syst. Signal Process.*, vol. 21, no. 5, pp. 2237–2247, 2007.

[17] A. S. Deepthi, “Anomaly Detection Using Principal Component Analysis 1 1 2,” *Int. J. Comput. Sci. Technol.*, vol. 8491, no. 4, pp. 124–126, 2014.

[18] M. Ding and H. Tian, “PCA-based network Traffic anomaly detection,” *Tsinghua Sci. Technol.*, vol. 21, no. 5, pp. 500–509, Oct. 2016.

[19] L. Huang, X. Nguyen, M. Garofalakis, M. I. Jordan, A. Joseph, and N. Taft, “In-Network PCA and Anomaly Detection,” *Adv. Neural Inf. Process. Syst. 19*, vol. 19, no. July 2014, pp. 617–624, 2007.

[20] A. Lakhina, M. Crovella, and C. Diot, “Diagnosing network-wide traffic anomalies,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, p. 219, 2004.

[21] V.-H. Le and S.-R. Kim, “K-strings algorithm, a new approach based on Kmeans,” 2015, pp. 15–20.