sensors

MDPI

Article

On the Search for Potentially Anomalous Traces of Cosmic Ray Particles in Images Acquired by Cmos Detectors for a Continuous Stream of Emerging Observational Data

Marcin Piekarczyk * and Tomasz Hachaj

Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering, AGH University of Krakow, Al. Mickiewicza 30, 30-059 Krakow, Poland; thachaj@agh.edu.pl
* Correspondence: mpiekarczyk@agh.edu.pl

Abstract: In this paper we propose the method for detecting potential anomalous cosmic ray particle tracks in big data image dataset acquired by Complementary Metal-Oxide-Semiconductors (CMOS). Those sensors are part of scientific infrastructure of Cosmic Ray Extremely Distributed Observatory (CREDO). The use of Incremental PCA (Principal Components Analysis) allowed approximation of loadings which might be updated at runtime. Incremental PCA with Sequential Karhunen-Loeve Transform results with almost identical embedding as basic PCA. Depending on image preprocessing method the weighted distance between coordinate frame and its approximation was at the level from 0.01 to 0.02 radian for batches with size of 10,000 images. This significantly reduces the necessary calculations in terms of memory complexity so that our method can be used for big data. The use of intuitive parameters of the potential anomalies detection algorithm based on object density in embedding space makes our method intuitive to use. The sets of anomalies returned by our proposed algorithm do not contain any typical morphologies of particle tracks shapes. Thus, one can conclude that our proposed method effectively filter-off typical (in terms of analysis of variance) shapes of particle tracks by searching for those that can be treated as significantly different from the others in the dataset. We also proposed method that can be used to find similar objects, which gives it the potential, for example, to be used in minimal distance-based classification and CREDO image database querying. The proposed algorithm was tested on more than half a million (570,000+) images that contains various morphologies of cosmic particle tracks. To our knowledge, this is the first study of this kind based on data collected using a distributed network of CMOS sensors embedded in the cell phones of participants collaborating within the citizen science paradigm.

Keywords: high-energy particles; image-based detection; anomalies detectionl; principal components analysis; image processing; sequential karhunen-loeve transform; big data; citizen science

Citation: Piekarczyk, M.; Hachaj, T. On the Search for Potentially Anomalous Traces of Cosmic Ray Particles in Images Acquired by Cmos Detectors for a Continuous Stream of Emerging Observational Data. *Sensors* **2024**, *24*, 1835. https://doi.org/10.3390/s24061835

Academic Editor: Alfio Dario Grasso

Received: 5 January 2024
Revised: 11 February 2024
Accepted: 7 March 2024
Published: 13 March 2024

1. Introduction

The problem of automatic anomaly detection is seen as one of the significant challenges in the analysis and recognition of measurement data. Anomaly detection concerns the search for those observations that deviate from the definition of normality for the considered set of observations [1,2]. Sometimes, interchangeable terms such as outlier detection or novelty detection are also used in this context, although they are not necessarily completely analogous [3–7]. This area has been actively developed in recent years, and many methods have been proposed in this field of research [8,9]. Among the first techniques proposed to deal with anomalies detection were statistical methods [10,11], especially those related to density estimation like KDE (Kernel Density Estimation) [12]. Nowadays, many solutions apply various machine learning methods, like shallow and deep models [5,13,14].

Anomaly detection techniques are applied to analyze and solve a wide range of problems in various areas. Examples of practical applications include cybersecurity (intrusion detection systems) [15–19], economy and healthcare (fraud detection) [20–25], industry

(fault diagnosis, damage detection) [26–31], medicine (medical diagnosis, disease outbreak detection) [32–36], earth sciences (event detection) [37–40], bioinformatics [36,41–43], genetics [44,45], physics [46–50] or astronomy [51–56].

The ability to detect non-trivial observations that deviate from a consistent data stream is a particularly big challenge in particle physics and astronomy [57]. The search for unusual data can lead to the discovery of unknown physical phenomena [58–60]. Creating effective tools for such type of automatic analysis and identification is a very important research subject in particle physics and astronomy.

Research in particle physics is performed on data acquired from experiments performed on large-scale stationary particle accelerators at projects such as LHC/CERN (Large Hadron Collider) [61–64], SLAC (Stanford Linear Accelerator Center) [65], Thomas Jefferson National Accelerator Facility [66,67], J-PARC (Japan Proton Accelerator Research Complex) [68,69] and many others. There are also large-scale observatories that measure cosmic radiation arriving from space. Among them are Pierre Auger Observatory [70,71], IceCube [72] or Telescope Array Project [73]. Stationary observatories of this type perform very accurate measurements. However, the observations they make are limited to the area where their research infrastructure is located. Due to this fact, they observe only a certain fraction of cosmic radiation reaching the Earth's atmosphere. To overcome the limitations of stationary observatories, several projects have been developed in recent years that allow distributed observations of cosmic radiation. These projects are based on the citizen science paradigm and use CMOS/CCD camera-based particle detectors [74]. Projects of this kind are:CRAYFIS (Cosmic RAYs Found In Smartphones) [75,76], DECO (Distributed Electronic Cosmic-ray Observatory) [77,78] and CREDO [79,80]. CRAYFIS [81] is a globally distributed network of cosmic-ray sensors for the exploration of cosmic rays, with the potential to reveal unexpected or previously unobserved planet-scale phenomena such as widely separated simultaneous extensive air showers. DECO is a similar project which utilizes smartphone-based cosmic rays detectors. The project is conducting advanced research on detection and classification of particle types based on deep learning models [82]. CREDO project uses smartphone-based detectors and additionally integrates data from other sources such as simple scintillation detectors [83–87]. The data collected by the CREDO project is stored in open repositories and are available for scientific purposes. In all of these projects, the use of mobile detectors based on optical sensors that record traces of particle radiation energy offers great flexibility and the possibility to extend observation coverage on a global scale [81]. Detectors acquires a huge amount of measurement data of various types, which requires appropriate analysis especially automatic recognition in big data streams [88,89].

The main purpose of searching for unusual signals (anomalies) in such data sets is to look for new physical phenomena (unknown physics) [79]. Such new phenomena might be potentially registered as non-typical particle traces observed on detector arrays and might be evidence of new particles or physical interactions. Such phenomena can occur when ultra high-energy cosmic radiation strikes the Earth and creates a stream of secondary particles observed by detectors. It should be noted that the primary particles hitting the atmosphere can have energies far beyond the energy ranges achievable in Earth's laboratories, creating unique physical conditions. Detection of unusual particle images observed on a globally distributed set of detectors also has the potential to reveal unexpected or previously unobserved phenomena occurring at the planetary scale [81]. Such phenomena might be revealed for example if similar types of anomalies occur in remote geographic locations corresponding to independent or simultaneous extensive air showers (EAS). Statistical analysis of anomalies in a large dataset is also a useful tool for tuning detection and filtering algorithms for observed events. Such analysis also makes it possible to study the response of a variety of CMOS sensors to radiation by analysis of a statistically significant number of actual measurements.

The problem addressed in this paper concerns the detection of anomalies in CREDO data acquired from smartphone-based mobile detectors. The primary carrier of information

in this case are images of particle tracks recorded on CMOS arrays [74,90]. Since the data is collected in continuous mode, it is necessary to take into account the possibility of streaming digging through the dataset for unusual observations. So far CREDO data has been analyzed for both background signal filtering and artifacts removal [91–93], as well as classification and recognition [92,94–96]. There have also been initial works on detecting abnormal data based on various techniques such as rough sets [97].

1.1. Novelty of This Research

To our knowledge, this is the first study which proposes a method that can detect potential anomalies in a continuous data stream and find objects with similar morphological structure in cosmic rays unlabeled data collected by a distributed network of CMOS sensors embedded in the cell phones. The proposed solution has been implemented and validated on the largest dataset of its kind to date, containing over 570,000 images. An important fact is that our approach has no limitations due to the size of the dataset, as embedding can be calculated and updated relatively quickly using small batches of new data. We were able to achieve this by using incremental PCA (Principal Components Analysis) feature extractions [98–101], appropriate image preprocessing and density-based anomalies search. In practice, the method presented in this paper has the potential for immediate detection of potential anomalies in the data stream incoming from the entire CREDO observatory network.

1.2. Paper Structure

The rest of the article is organized as follows. Section 2 discusses the structure of the CREDO data subset used in the article, explains the preprocessing of the raw image data, the mathematical basis of Incremental PCA, and the scheme of the anomaly detection algorithm. We have divided the presentation and discussion of the results into two sections. Section 3 presents technical aspects of the proposed method. Section 4 contains detailed discussion and interpretation of results. The Section 5 summarizes the scientific contributions.

2. Material and Methods

Since the research problem addressed in this article concerns the search for anomalies in CREDO imaging data, it is necessary to start by defining how we understand these anomalies. Due to the nature of the observations, we are dealing with traces left by energy-carrying particles on a CMOS array. From a physics point of view high-energy particles should left traces in the shape of low-density point or thin lines [80]. Observations of this shape are the majority of the dataset. Patterns that deviate significantly from these standards can be treated as anomalies. We are unable to identify a reference pattern for anomalies, as they can have morphologically very different shapes.

2.1. Datasets

The CREDO dataset is currently the largest open dataset containing recorded traces of potential cosmic ray particles acquired by mobile detectors. To our knowledge, there is no other such comprehensive dataset of this modality, which is additionally constantly updated with new recorded events. For this reason, we applied it to our research as a state-of-the-art data repository in the field of citizen science-based cosmic rays observations. A subset of CREDO data from Android-based mobile detectors was used to verify the solutions proposed in the article. It consists a set of observations saved in digital images with resolution 60×60. Each recorded observation is also associated with metadata such as acquisition time, geographic coordinates, etc. Those additional information is not considered in the algorithms presented. The data was recorded during 2023 year and passed the standard anti-artifact filter used in the project [74]. The size of the dataset we used in this research is 573,335 images.

2.2. Image Prepossessing (Aligning)

Image aligning might improve results of further image analysis [102–105]. In case of CREDO dataset, the aligning is based on translating images so that the pixels with the highest grayscale intensity will be in the center of the image, and rotating images so that the brightest collinear pixels will be horizontal. This type of alignment might be done with the aid of PCA. The proposed aligning algorithm works as follows:

- Input image is converted to grayscale;
- PCA is computed on a dataset constructed from pixels of grayscale image. Each pixel has its coordinate in the image. If the pixel is black (has value equals 0) its coordinates are not included in the dataset. If the pixel has a value greater than zero, we add to the dataset as many points with coordinates of that pixel as the value of that pixel (from 1 to 255). This means that the brighter the pixel is, the more data it appends to the dataset from which the PCA is calculated;
- Most significant PCA axis is used to rotate image while dataset mean is used to translate image;
- After image rotation and translation result image is cropped to original size of input image. Due to this fact some pixels in image borders might not have calculated pixels value. In order to calculate those border pixels we perform pixel extrapolation. We have used following pixel extrapolation methods which are defined in OpenCV [106] (see Table 1).
  - B. Constant–no matter of image colors "abcd", border (not defined by transform) pixels are assigned to have constants color "o".
  - B. Reflect–border pixels (not defined by transform) are reflections of image colors. For example if image colors are "abcd" left border will have extrapolated values "...dcb" and right border will have extrapolated values "cba...".
  - B. Replicate–border pixels (not defined by transform) are the same pixels that are positioned on the edge of image which has pixels defined by a transform. For example if image colors are "abcd" left border will have extrapolated values "...aaa" and right border will have extrapolated values "ddd...".

The images are converted to grayscale as part of processing, however in all figures we present original images in RGB color scale.

The proposed algorithm pseudo code is presented in Algorithm 1. In Table 1 we present image aligning methods we used during dataset preprocessing. We have tested four methods: no preprocessing (None) which use raw data and Algorithm 1 with all three extrapolation methods we have described above.

Table 1. Image aligning methods we used during dataset preprocessing. Column titled "Aligning and pixel extrapolation with example" gives example results of pixel extrapolation algorithm for borders.

Image Alignment	Aligning and Pixel Extrapolation with Example
None	Algorithm 1 is not applied (further processing of rough data)
B. Constant	Algorithm 1, ooo | abcd | ooo with specified o
B. Reflect	Algorithm 1, dcb | abcd | cba
B. Replicate	Algorithm 1, aaa | abcd | ddd

2.3. PCA-Based Features

Principal components analysis is a statistical method based on covariance analysis that finds the transformation matrix which allows projecting the dataset to lower dimension with linear transform that preserves maximal number of information in the sense of preserving variance. In other words object in a dataset can be described with fewer dimension than with initial one. There are several methods that can be used in place of PCA for feature extractions. Popular methods of this type include Independent component analysis (ICA) [107] or utilizing the latent space from various Encoder-Decoder deep neural networks (E-D) [108]. Among the most important limitations of PCA are the facts

that it is only a linear transformation, as well as usually requires scaling of individual features. In our case, feature scaling is not necessary because we are dealing with image files where the signal representation is limited and quantized. Unlike ICA, PCA does not require that the signals we want to extract meet the assumptions of independence, having non-Gaussian histograms, and having lower complexity than mixture signals. PCA also has some advantages over the E-D approach. PCA allows exact calculation of information loss due to dimensionality reduction, since in PCA one can easily estimate the percentage of variance explained by a subset of the selected coordinate system axes. Thanks to that, one can control the size of PCA embedding without the necessity of recalculating the projection matrix. In case of E-D the latent space is derived from the network's bottleneck and its size cannot be modified without retraining the whole architecture.

Algorithm 1: Image aligning PCA-based algorithm [109]

Data: Input: I image to be aligned, *mode*–pixel extrapolation algorithm
Result: I_a aligned image with accumulated pixels intensity centered and directed horizontally.

```
// Initialize empty list of points coordinates
```
$\mathrm{Y} \leftarrow \varnothing$;
```
// Convert input image to grayscale using weighted formula
   (see [109])
```
$I_g \leftarrow grayscale(I)$;
```
// For each pixel in image I_g
```
for $p \in \mathcal{I}_g$ **do**
    $a \leftarrow 0$;
    // Add coordinates of non-zero pixels so many times as the value
       of pixels is
    while $a < p.value$ **do**
        $\mathrm{Y} \leftarrow \mathrm{Y} \cup p.xy$;
        $a \leftarrow a + 1$;
    end
end
```
// Perform PCA on dataset in list Y, calculate eigenvectors [2 × 2]
   matrix V and eigenvalues vector $\overline{\lambda}$ (there are two eigenvalues) and
   vector with coordinate of mean value $\overline{m}$
```
$V, \overline{\lambda}, \overline{m} \leftarrow PCA(\mathrm{Y})$;
```
// Calculate rotation of principal axis
```
$\alpha \leftarrow atan2(V_{[1,2]}, V_{[1,1]})$;
```
// Calculate rotation and translation matrix A
```
$$A \leftarrow \begin{bmatrix} cos(\alpha) & -sin(\alpha) & \overline{m_1} \cdot (1 - cos(\alpha)) + \overline{m_2} \cdot sin(\alpha) \\ sin(\alpha) & cos(\alpha) & \overline{m_2} \cdot (1 - cos(\alpha)) - \overline{m_1} \cdot sin(\alpha) \end{bmatrix};$$
```
// Multiply coordinate of each pixel of I_g by A to get new
   coordinates
```
$I_a \leftarrow A \times I_g$;
```
// Crop image to initial (rectangular) shape of I_g and extrapolate
   pixels colors in regions, that do not have values
```
$I_a \leftarrow crop\_interpolate(I_a, mode)$;
return I_a

Let the dataset Ψ contains n images. Each image I_i has resolution $d_1 \times d_2$, $i \in [1 \dots n]$. Values of pixels in image are in range $[0, 1]$. For the rest of the paper let us consider two-dimensional image as one dimensional vector of length $d = d_1 \cdot d_2$. To make this 2d to 1d conversion we store each row of the image one by another in a single row vector. In order to calculate PCA-based embedding (we will call it also "basic PCA") of image we can adapt

eigenfaces-based image representation similar to one used in [110]. In order to do so we need to create a matrix that contains all observations:

$$X \leftarrow= [I_1, \ldots, I_n]^T \tag{1}$$

where X has $n \times d$ dimensions. The number of rows is the same as the number of images in the dataset while number of columns is the same as images dimension. The next step is to calculate mean vector of each column of the matrix X, $mean_{col}(X)$ and then covariance matrix:

$$T = \frac{(X - mean_{col}(X))^T \times (X - mean_{col}(X))}{d} \tag{2}$$

where T is a square $d \times d$ matrix.

Matrix T is then a subject of eigendecomposition in order to find set of eigenvectors stored in matrix V and corresponding to them eigenvalues $\overrightarrow{\lambda}$. Let us assume that PCA loadings are positioned in columns. After ordering eigenvectors in order of descending absolute values of eigenvalues the embedding E is calculated according to equation:

$$E = (V \times (X - mean_{col}(X))^T)^T \tag{3}$$

where E is a $n \times d$ dimensional embedding. In order to perform dimensional reduction, we need to skip certain rows in matrix V, for example when we leave only 5 first rows the embedding will be 5 dimensional and matrix E will become $n \times 5$.

2.4. Potential Anomalies Detection

After applying PCA and dimensions reduction we can use a new obtained embedding (latent) space to examine similarity between objects. We can define an anomaly as an object that is not similar to other objects in the dataset in terms of distance between objects embedding. According to this definition outliers might be considered as anomalies. In order to detect outliers we can apply certain cluster analysis algorithm like agglomerative clustering [111], DBSCAN [112] or even k-means [97]. In case of first two algorithms in order to optimize performance it is required to calculate distance matrix between objects in the dataset which might be difficult or hardly possible in case of big data. In our case we do not need to find the answer to which cluster a certain objects belongs, rather if a certain object is outlier. Knowing this we can adapt the anomalies searching approach derived from the DBSCAN: an object I_j with embedding E_j is an outlier when in its neighbourhood with radius ϵ there are less than k other objects. The anomalies set A can be defined as:

$$I_j \in A \Leftrightarrow \#\{I_i : d(E_j, E_i) < \epsilon, i \in [1 \ldots n]\} < k \tag{4}$$

where # is cardinal number of the set and $\{I_i : d(E_j, E_i) < \epsilon, i \in [1 \ldots n]\}$ is a set of objects which distances between their embedding and embedding of I_j is less than ϵ and d is a distance function (in our case Euclidean distance).

The algorithm that detect potential anomalies according to Equation (4) has complexity $O(n) = n^2$ however because evaluation of each object I_j in the dataset is independent of the others it can be easily speed up by the map-reduce approach on the parallel processing pipeline.

2.5. Querying the Object Database for the Most Similar Objects

The procedure of finding k most similar objects to I_j requires calculating distance between embedding of this object and embedding of each other object and ordering them in descending order. Objects corresponding to first k smallest distances indicate most similar objects.

$$D_ist(E_j) = [(d(E_1, E_j), E_1), \ldots, (d(E_i, E_j), E_i), \ldots, (d(E_n, E_j), E_n)] \text{ ordered by } d(E_i, E_j) \tag{5}$$

where $D_ist(E_j)$ is ordered list of pairs, each pair contains distance between E_j and certain element from the dataset Ψ. Pairs in list are ordered by descending order by calculated distance, $i \in [1 \dots n]/j$.

Formally set of k most similar objects is defined as:

$$Sim(I_j, k) = \{I_i : position(E_i)\ in\ D_{ist}(E_j) \leqslant k\} \tag{6}$$

where $position(E_i)$ returns index of element E_i in ordered list $D_{ist}(E_j)$.

According to Equation (6) if two or more objects have the same distance to E_j it is possible that more than k objects will be returned.

Searching the image database for the k most similar objects to I_j thus reduces to finding all I_i that satisfy (6). If Algorithm 1 for image aligning and (3) or Algorithm 2 for embedding calculation is applied, the search process reduces to the pairwise distance calculation problem.

2.6. Approximation of PCA for Big Data

The calculation of PCA features with an algorithm given in the Section 2.3 has a memory and computational bottleneck when the covariance matrix is calculated according to Equation (2). The rest of the computation is done on a fixed-size matrix. The matrix X (see (1)) occupies $n \cdot d_1 \cdot d_2 \cdot bc$ in memory, where bc is the number of bytes allocated to represent the floating-point number. For the real world data considered in this work, in the case of large image datasets, for example, with quantity of 10^6 images and a resolution of 60×60 pixels, the matrix T stored with double precision (8 Bytes) occupies in memory: $10^6 \cdot 60 \cdot 60 \cdot 8 \approx 26.8$ GB and grows linearly as the number of images in the dataset increase. In order to reduce the memory and computational complexity of the algorithm finding image embedding, one can use PCA approximation based on incremental calculation of PCA with, for example, the algorithm proposed in the paper [101]. That algorithm is an extension of the Sequential Karhunen-Loeve Transform [113]. A mean update is calculated according to a Youngs and Cramer variance update procedure [114]. The method is called Incremental PCA and works as described in Algorithm 2. Returned matrices V_a^T and S_a being approximations of PCA can be used in (3) to calculate embedding.

2.7. Detecting Potential Anomalies in Big Dataset under Condition of Continuously Incoming Objects

To perform anomaly detection on the dataset described in Section 2.1, one needs to do image aligning using Algorithm 1, generate an embedding of the dataset using Equation (3) and then use (4) at a fixed (ϵ, k). However, this approach requires calculating the memory-expensive Equation (2). If cosmic ray particle images are acquired continuously, the Equation (2) will have to be repeated from time to time, for example, when a new large enough batch of data is collected. We cannot assume that the dataset we have gathered so far is representative and skip updating (3), because new devices may be incorporated into the CREDO sensor network, from which the resulting data will have different characteristic from those acquired earlier. This will also requires updating the statistical parameters obtained from the PCA. In order to reduce the number of necessary calculations, the step of determining PCA with a basic algorithm, for example, based on SVD (Singular value decomposition), can be replaced by approximation of PCA by Algorithm 2. Algorithm 2 will be run every time a batch of new data of the certain size is collected. The rest of the data processing pipeline will look identical like in the case with the basic PCA. Note that (3) can be performed iteratively for individual images or groups of images, not necessarily for all of X at once. Anomalies detection with (4) can be run after each update of embedding, either for whole X or only for new objects in batch. The procedure depends on the strategy adopted, for example, whether one wants to repeatedly analyze the same (old) data in search of potential anomalies. Evaluation of the dependence of the obtained embedding on the size of the training dataset and the differences in the found anomalies will be analyzed in the following sections.

Algorithm 2: Incremental PCA algorithm

Data: Input: X is two-dimensional data array with n rows and d columns; bs–batch size.

Result: V_a^T is $d \times d$ matrix with approximation of PCA loadings; S_a is $d \times d$ diagonal matrix with eigenvalues corresponding to eigenvectors in V_a^T.

```
// Number of samples from X analyzed so far
```
$nsa \leftarrow 0$;
```
// Approximation of mean value of matrix X columns
```
$\overrightarrow{mean} \leftarrow \overrightarrow{0}$;
$a \leftarrow 0$;
while $nsa + bs < n$ **do**
```
    // Get bs rows from matrix X and store them in matrix X'
```
    $X^{'} \leftarrow X[a : (a + bs)]$;
    $\overrightarrow{ls} \leftarrow \overrightarrow{mean} \cdot nsa$;
```
    // Sum values in each column of X' and store it in vector ns
```
    $\overrightarrow{ns} \leftarrow columnsum(X^{'})$;
    $nts \leftarrow nsa + bs$;
```
    // Element-wise multiplication
```
    $\overrightarrow{colmean} \leftarrow \frac{\overrightarrow{ls} + \overrightarrow{ns}}{nts}$;
    if $nsa = 0$ **then**
```
        // First iteration
        // From each column of X' subtract vector colmean
```
        $X^{'} \leftarrow X^{'} - \overrightarrow{colmean}$;
    end
    else
```
        // Next iterations
        // Calculate mean value of each column of X' and store it in
           vector mean_cb
```
        $mean_{cb} \leftarrow columnmean(X^{'})$;
```
        // From each column of X subtract vector mean_cb
```
        $X^{'} \leftarrow X^{'} - \overrightarrow{mean_{cb}}$;
        $\overrightarrow{mean_{cor}} \leftarrow \sqrt{\frac{nsa}{nts}} \cdot bs \cdot (\overrightarrow{mean} - \overrightarrow{mean_{cb}})$;
```
        // Create matrix X' with d + bs + 1 rows and d columns, each
           matrix vector is stacked row after row
```
        $X^{'} \leftarrow \begin{bmatrix} S_a \cdot V_a^T \\ X^{'} \\ \overrightarrow{mean_{cor}} \end{bmatrix}$
    end
    ;
```
    // Calculate singular values decomposition of X'
```
    $[U_a, S_a, V_a^T] \leftarrow SVD(X^{'})$;
```
    // Sign correction is applied so that the rows in V that are
       largest in absolute value are always positive
```
    $V_a^T \leftarrow sc(V_a^T)$;
    $nsa \leftarrow nts$;
    $\overrightarrow{mean} \leftarrow \overrightarrow{colmean}$;
end
return $[V_a^T, S_a]$

3. Results

We implemented our solution using Python 3.8. The source code of the proposed algorithm and the dataset can be downloaded from GitHub repository https://github.com/browarsoftware/anomalies_bigdata (accessed on 20 December 2023). We have used numba 0.5, numpy 1.22, opencv-python 4.5, scikit-learn 1.0, scipy 1.8 Python libraries. Plots were made in R langauge 3.6.

The purpose of the evaluation was to test the effectiveness of detecting potential anomalies using image preprocessing (aligning) methods described in Section 2.2, PCA-based features described in Section 2.3 and anomalies detection approach in Section 2.4.

The dataset presented in Section 2.1 was set randomly. In Table 2 we present a comparison of the resulting coordinate frames computed using basic PCA for four different preprocessing algorithms. The comparison of axes is intended to numerically calculate the difference between the potential embeddings and to indicate the effect of using different preprocessing methods on the calculation of PCA. The comparison of coordinate systems is done using coordinate frames weighted distance (cfd):

$$cfd((V_1, \overline{\lambda_1}), (V_2, \overline{\lambda_2})) = \sum_{i=1}^{d} \frac{\overline{\lambda_{1,i}} + \overline{\lambda_{2,i}}}{2} \measuredangle(sc(V_{1,i}), sc(V_{2,i})) \tag{7}$$

where $(V_1, \overline{\lambda_1})$ is eigenvectors matrix and eigenvalues vector of first PCA, $(V_2, \overline{\lambda_2})$ is eigenvectors matrix and eigenvalues vector of second PCA, $(V_{1,i}, \lambda_{1,i})$ is i-th eigenvector and i-th eigenvalue of first PCA, sc is a sign correction (see Algorithm 2) and $\measuredangle$ is an operator for calculating the angle between vectors. Note that all eigenvalues of PCA are non-negative; cfd is measured in radians (rad).

Table 2. A comparison of the resulting coordinate frames computed using basic PCA for four different preprocessing algorithms. The comparison of coordinate systems is done using coordinate frames weighted distance (cfd) and it is measured in radians.

	None	**B. Constant**	**B. Reflect**	**B. Replicate**
None	0	0.531	0.176	0.178
B. Constant	0.531	0	0.520	0.522
B. Reflect	0.176	0.520	0	0.042
B. Replicate	0.178	0.522	0.042	0

To perform embedding we used 62 features out of 3600 that is, we reduced the dimensionality of embedding to 62 dimensions. Such a reduction explains, depending on the preprocessing method adopted, between 98% and 99% of the total variance in our dataset. We decided to adopt such a number of dimensions because it allowed us to more easily manipulate the value of ϵ in (4), which must be determined depending on the number of dimensions and in practice cannot be determined other way than experimentally, as in DBSCAN algorithm.

We made a comparison of the sets of potential anomalies returned by the method described by Equation (4) for different preprocessing algorithms from Section 2.2 and embedding calculated with basic PCA algorithm from Section 2.3. We used Jaccard index (J) [115] and Overlap coefficient (OC) [116] to compare the sets of anomalies:

$$J(A_1, A_2) = \frac{A_1 \cap A_2}{A_1 \cup A_2} \tag{8}$$

where A_1, A_2 are potential anomalies sets to be compared.

$$OC(A_1, A_2) = \frac{A_1 \cap A_2}{min(\#A_1, \#A_2)} \tag{9}$$

Figures 1 and 2 present comparison of results of potential anomalies detection with (4) evaluated with (8) and (9). Types of preprocessing and values of ϵ are in Table 3. The k parameter in (4) was arbitrarily set to 3. The first sixteen potential anomalies for each of the four preprocessing methods calculated for basic PCA with $(\alpha = 2.4, k = 3)$ are shown in Figure 3.

Table 3. Description of Algorithms in Figures 1 and 2. Columns show algorithm id, type of preprocessing and value of ϵ in (4).

Algorithm id	Image Alignment	α
1	None	3.0
2	None	2.8
3	None	2.6
4	None	2.4
5	B. Constant	3.0
6	B. Constant	2.8
7	B. Constant	2.6
8	B. Constant	2.4
9	B. Replicate	3.0
10	B. Replicate	2.8
11	B. Replicate	2.6
12	B. Replicate	2.4
13	B. Reflect	3.0
14	B. Reflect	2.8
15	B. Reflect	2.6
16	B. Reflect	2.4

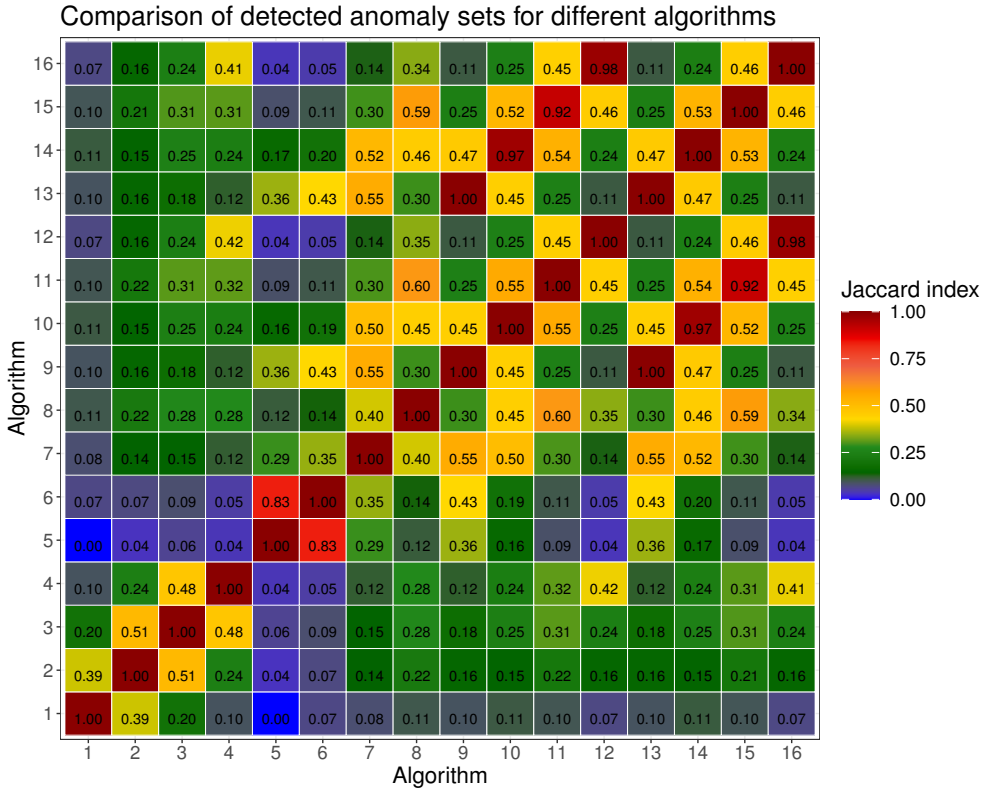

Figure 1. Comparison of results of potential anomalies detection with (4) evaluated with Jaccard index (8). Types of preprocessing and values of ϵ are in Table 3. The k parameter in (4) was arbitrarily set to 3.

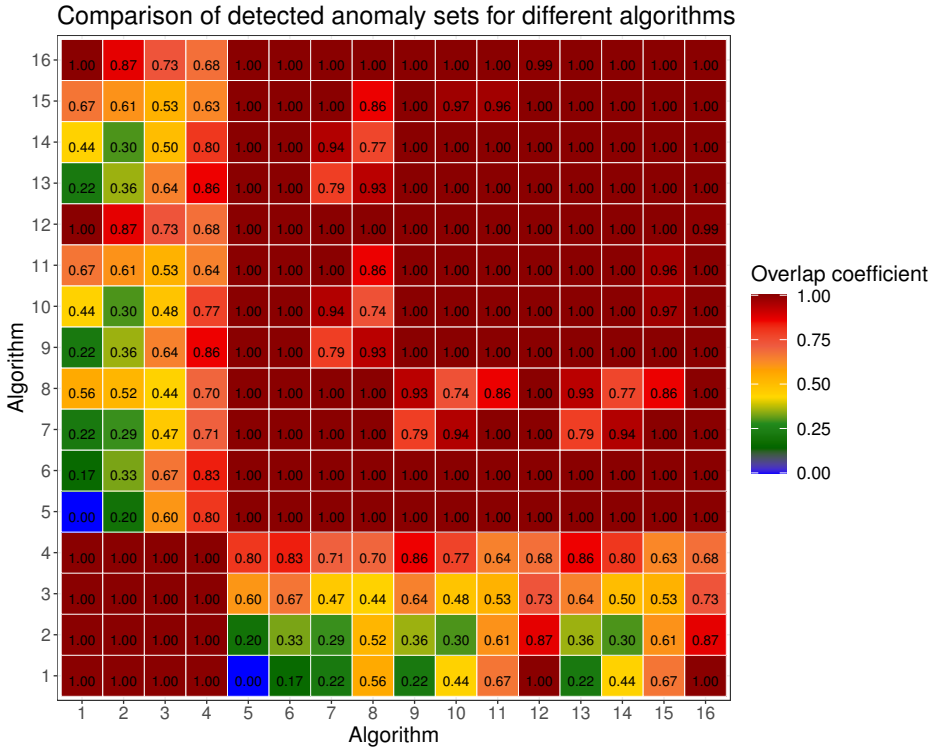

Figure 2. Comparison of results of potential anomalies detection with (4) evaluated with Overlap coefficient (9). Types of preprocessing and values of ϵ are in Table 3. The k parameter in (4) was arbitrarily set to 3.

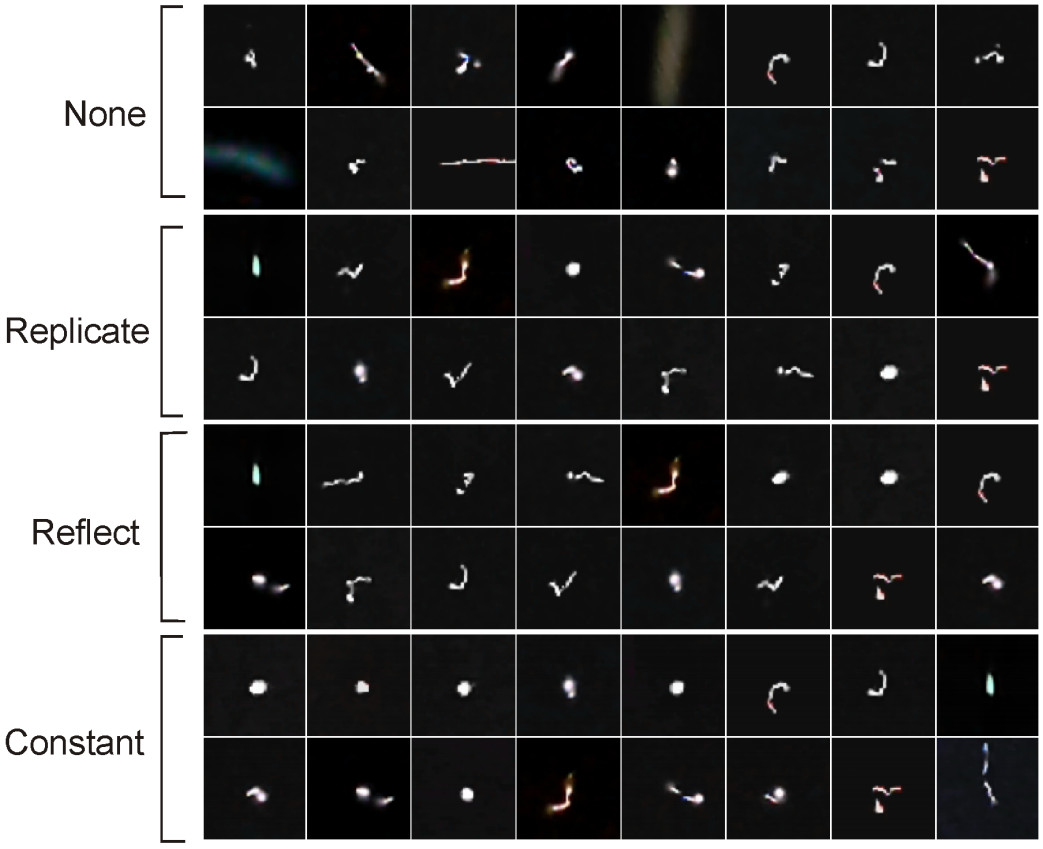

Figure 3. The first sixteen potential anomalies for each of the four preprocessing methods calculated for basic PCA with $(\alpha = 2.4, k = 3)$ in (4).

The next stage of the evaluation was to test the effectiveness of using Incremental PCA (see Section 2.6) in the procedure for detection of potential anomalies in dateset under condition of continuously incoming data (see Section 2.7). In order to do so, we made a comparison of coordinate frames obtained with basic PCA to coordinate frames obtained with Incremental PCA for a different number of data used when approximating PCA with Algorithm 2. The results are shown in Figure 4. Each point on the plot shows the cfd value (7) for the PCA coordinate axes calculated on the whole data and the coordinate axes calculated by Incremental PCA on a certain percentage of the whole data, that is, for example, the PCA axes calculated on the whole set with B. Replicate preprocessing and the axes calculated with Incremental PCA with B. Replicate preprocessing calculated on 10%, 20%, 30% of the data etc. We presented the selected cfd values for this evaluation in Table 4. For Incremental PCA, we assumed a batch size (bs) of 10,000.

Table 4. Coordinate frames weighted distance between PCA and Incremental PCA. Weighted distance is expressed in radians [rad]. Each batch contains 10^4 images.

Batch Number	None	B. Constant	B. Reflect	B. Replicate
1	0.107	0.107	0.115	0.133
3	0.089	0.093	0.102	0.117
5	0.078	0.107	0.074	0.104
7	0.072	0.068	0.059	0.098
9	0.071	0.070	0.047	0.106
11	0.072	0.067	0.051	0.100
13	0.081	0.071	0.052	0.096
15	0.067	0.066	0.049	0.091
17	0.063	0.058	0.047	0.080
19	0.062	0.051	0.040	0.078
21	0.064	0.060	0.041	0.075
23	0.057	0.056	0.042	0.075
25	0.054	0.048	0.041	0.057
27	0.053	0.041	0.038	0.058
29	0.058	0.040	0.040	0.059
31	0.051	0.040	0.037	0.069
33	0.051	0.039	0.044	0.070
35	0.050	0.040	0.037	0.062
37	0.049	0.039	0.042	0.059
39	0.048	0.042	0.044	0.063
41	0.047	0.038	0.041	0.055
43	0.047	0.035	0.039	0.055
45	0.048	0.031	0.035	0.055
47	0.047	0.029	0.036	0.060
49	0.045	0.026	0.032	0.050
51	0.043	0.025	0.032	0.036
53	0.035	0.023	0.025	0.042
55	0.031	0.019	0.024	0.026
57	0.020	0.010	0.012	0.013

Then we made a comparison of Jaccard Index and Overlap Coefficient of the method for finding potential anomalies (4) with the parameters ($\alpha = 2.4$, $k = 3$) for PCA and Incremental PCA calculated on increasing numbers of data. Since the results for each image alignment were very similar on Figure 5 we present the results for B. Reflect only. We always performed embedding on the entire dataset and we calculated Incremental PCA for some subset of the data, thus simulating a constant increment of the data on which embedding is performed relative to the data used when counting embedding. The number of data used by Incremental PCA is coded in Figure 5 as follows:

1. basic PCA (calculated on full dataset),
2. Incremental PCA calculated on $56 \cdot 10^4$ images.
3. Incremental PCA calculated on $46 \cdot 10^4$ images,

4. Incremental PCA calculated on $36 \cdot 10^4$ images,
5. Incremental PCA calculated on $26 \cdot 10^4$ images,
6. Incremental PCA calculated on $16 \cdot 10^4$ images,
7. Incremental PCA calculated on $6 \cdot 10^4$ images,

The last step of the evaluation is to check the effectiveness of the method that detects similar objects according to Equation (6). For this purpose, we used the preprocessing algorithm B. Reflect and we generated features using basic PCA. We do not present the results obtained with Incremental PCA because, as will be shown in the discussion, they are virtually identical to basic PCA. We selected 9 sample images representing characteristic shape morphologies of particle tracks in the dataset and found $k = 7$ most similar images according to Equation (6). We presented the results in Figure 6.

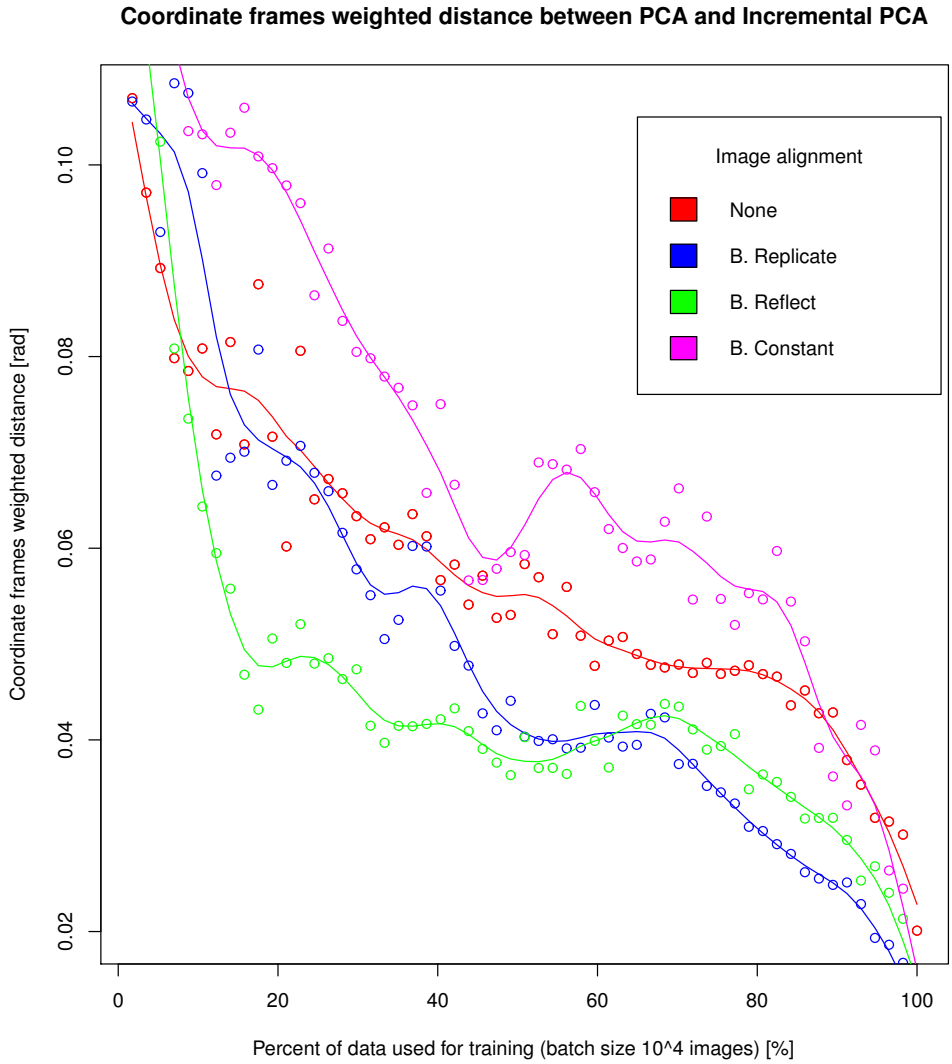

Figure 4. Comparison of coordinate frames obtained with basic PCA to coordinate frames obtained with Incremental PCA for a different number of data used when approximating PCA with Algorithm 2. Each point on the plot shows the cfd value (7) for the PCA coordinate axes calculated on the whole data and the coordinate axes calculated by Incremental PCA on a certain percentage of the whole data, that is, for example, the PCA axes calculated on the whole set with B. Replicate preprocessing and the axes calculated with Incremental PCA with B. Replicate preprocessing calculated on 10%, 20%, 30% of the data etc.

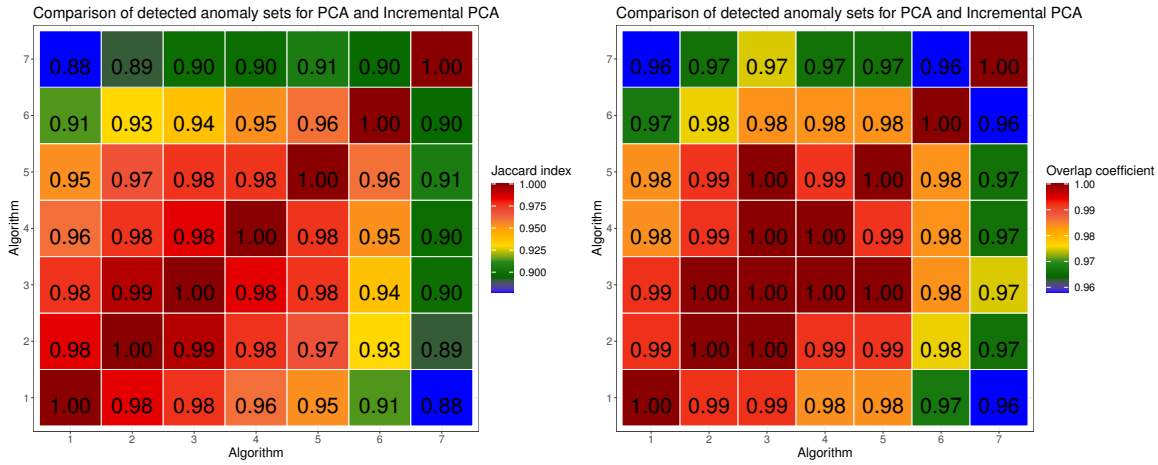

Figure 5. A comparison of Jaccard Index and Overlap Coefficient for the method of finding potential anomalies (4) with the parameters ($\alpha = 2.4$, $k = 3$) for PCA and Incremental PCA counted on increasing numbers of data. We performed embedding and potential anomalies detection on the entire dataset.

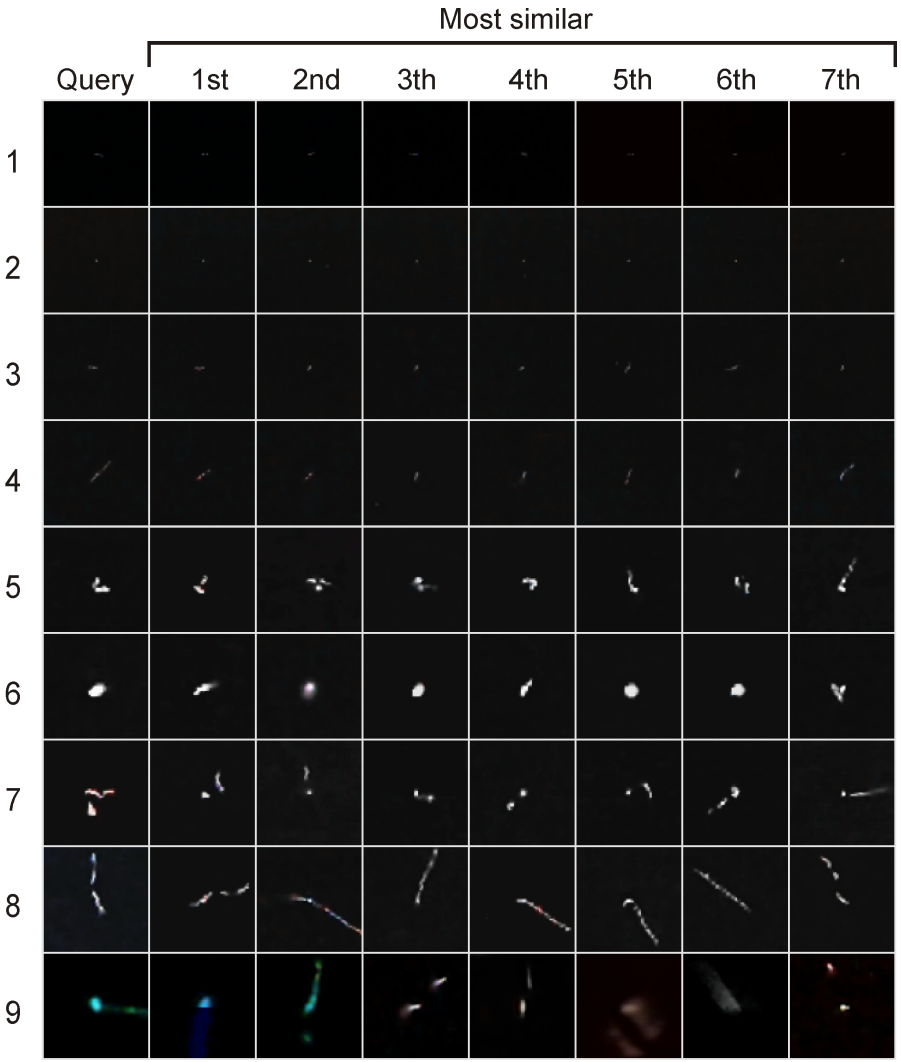

Figure 6. Test of the effectiveness of the method that detects similar objects according to Equation (6). For this purpose, we used the preprocessing algorithm B. Reflect and we generated features using basic PCA. The first column contains image I_j (see Equation (6)). Each subsequent column contain the most similar images, the further to the left the Euclidean distance between embedding E_i and E_j is higher (second from left is most similar to first, first from right is the least similar from all seven).

4. Discussion

Based on the results shown in Table 2, it can be concluded that the individual coordinate frames calculated on datasets with different embeddings differ from each other considering the cfd measure (7). In the case of lack of preprocessing (None) and B. Constant the differences between the obtained coordinate frames are the largest. This is due to the fact that B. Constant attaches black pixels on borders to the resulting image, which are not present in such numbers in raw images. There is a little difference between coordinate frames calculated on data processed with B. Reflect and B. Replicate, it amounts 0.042 rad. Although the difference is small, there is no guarantee that the embedding calculated with PCA on the set preprocessed with one method can be used interchangeably with the embedding calculated on the set preprocessed with another method. The choice of a particular preprocessing method determines the necessity of its use in subsequent stages of dataset analysis.

Although the different methods create different embedding of the images, the sets of anomalies they find are not significantly different. According to the results in Figures 1 and 2, the number of anomalies found naturally decreases as ϵ increases. This can be observed when comparing two anomalies detection methods with a larger and smaller ϵ value–the Jaccard Index has a smaller value when there is a larger difference of ϵ between those two methods. As expected for a certain preprocessing method, as the ϵ decreases, new objects are added to the set of anomalies without removing those found with a larger ϵ. This can be observed from the Overlap Coefficient, which always has a value of 1 within a single preprocessing method regardless of ϵ. It can also be seen from the Overlap Coefficient analysis that the use of a preprocessing method (other than None) results in each of the detection algorithms returning a very similar set of potential anomalies–OC equals almost always 1 and the smallest value is in the case of B. Constant $\epsilon = 2.4$ and B. Replicate $\epsilon = 2.8$ and equals 0.74. If we compare embedding based on preprocessing None with other methods Overlap Coefficient ranges from 0 to 1, which means that different sets of potential anomalies are returned. Thus, one can conclude that preprocessing affects the anomalies that we detect. In the case of the Jaccard Index, the values of this coefficient are in most cases less than 1. This means that for the same values of ϵ, the different preprocessing methods affect embedding in such a way that they search for sets of potential anomalies of different quantity. This confirms the results from Table 2 that the coordinate frames are different from each other and the distances between objects in the PCA-designated spaces are also different.

When designing potential anomaly search algorithm using PCA embedding, we do not define the particle trajectory morphologies of interest. We expect that if we do not apply preprocessing but work on embedding generated from raw dataset (in our case preprocessing equals to None), the returned potential anomalies are different from those found when we apply preprocessing. This expectation is confirmed in Figure 3. The set of the first 16 retrieved anomalies for each method at $\epsilon = 2.8$ in the case of preprocessing None returns a dataset different from those returned by Replicate, Reflect and Constant. However those three preprocessing methods returns very similar particle tracks. This means that moving and rotating the objects so that the largest variance of bright points is along the horizontal axis significantly affects the result. Basing on what we know about PCA in other image domains, it can be concluded that image alignment is a beneficial process for variance analysis. For this reason, we recommend the use image aligning. The sets of potential anomalies returned by proposed algorithm do not contain any typical morphologies of particle tracks shapes (see, for example, the results of Figure 3). Thus, one can conclude that our proposed method effectively filter-off typical (in terms of analysis of variance) shapes of particle tracks by searching for those that can be treated as significantly different from the others in the dataset.

Based on the results from Table 4 shown in Figure 4, it can be seen that as the number of data processed by Incremental PCA increases (in our case with batch size set to 10,000), the cfd between PCA approximation and basic PCA expressed in radians decreases. Already

for an approximately 40% of dataset, the difference between those two values is between 0.04 and 0.06 radians. The similarity of the coordinate frame calculated with basic PCA and Incremental PCA also affects the similarity of the obtained embedding and thus the detected anomalies. We performed such an analysis for preprocessing B. Reflect. As can be seen in Figure 5 for Incremental PCA calculated on 97% of data with batch size 10,000, $J = 98$, $OC = 0.99$, so the returned sets of potential anomalies are almost identical. As the data used to calculate Incremental PCA decreases, both coefficients also decrease but not significantly. For Incremental PCA calculated on 80% of data $J = 98$, $OC = 0.99$, for 62% $J = 96$, $OC = 0.98$, for 45% $J = 95$, $OC = 0.98$, for 28% $J = 91$, $OC = 0.97$, for 10% $J = 0.88$, $OC = 0.96$. This means that using Incremental PCA, which is recalculated with incoming data with a batch size of 10,000, we get almost identical anomalies detection results as for basic PCA. It can be concluded that our method, if the dataset is shuffled (and is representative) a small portion of the dataset used to calculate Incremental PCA can detect almost identical set of anomalies as calculated with basic PCA. The method we proposed in Section 2.7 for detecting of potential anomalies in large dataset under condition of continuously incoming objects works almost identically to the method using the entire dataset for PCA calculation. As a result, the approach we have proposed significantly reduces the memory and computational requirements of the algorithm for detecting anomalies and makes it possible to use it for big datasets.

Also the use of (6) to detect similar objects works as expected. It returns morphologically similar objects to the one being searched for. The results shown in Figure 6 for B. Reflect confirm that the returned objects I_i have a very similar shape to the searched image I_j. Thanks to using image aligning, the method is not sensitive to translation and rotation of objects in images. As can be seen, the method based on (6) handles well the morphology of dots, lines, worms and various types of complex shapes. When searching for similar objects, the method also returns objects with similar levels of background noise, which may not be entirely beneficial (compare first and second row in the Figure 6). At the moment, however, with the preprocessing method described by Algorithm 1 it is not possible to remove the background. This is a certain drawback of that method if it will be applied to search for morphologically similar objects not considering background. Despite this fact, its search results give very satisfactory results in terms of morphology search.

The proposed anomalies detection algorithm worked as expected. The images it found have anomalous features according to their definition (4), that is, they contain traces of potential particles whose morphology differs significantly from typical image classes, that is, dots, lines and worms. The use of PCA as a feature extraction method did not create concentric clusters of objects. Due to this fact, one cannot use distance-based measures to find the central object of potential clusters, e.g., the "most typical dot class trajectory" around which there are similar objects. This behavior was expected because PCA does not statistically differentiate the correct signal from background noise present in some images. For this reason, density-based clustering seems to be an appropriate approach for grouping objects with similar morphology. Because (4) defines anomalies using a density-based approach, it is impossible to say which potential particle trace is "more anomalous" than the other. By controlling the parameters (α, k) in (4) and using various types of preprocessing, we have the ability to search the entire dataset. As we indicated in Figures 1 and 2, the preprocessing method slightly affects the returned sets of anomalies.

We cannot exclude the possibility that some of these images are artifacts due to the access of visible light to the CMOS array. At this stage, we do not yet know the physical interpretations of the anomalies we are detecting. The main goal of our study was to create a method that would allow us to find them efficiently in large data sets. The physical interpretation of the results obtained is beyond the scope of this work and requires further research. Our proposed method is intended to be a useful mathematical tool for defining and finding potential anomalies.

In Figure 7 we present examples of anomalies detected by the proposed method with parameters $(\alpha = 2.3, k = 5)$, B. Replicate preprocessing, basic PCA. We chose them because

they represent a variety of deviations from the typical shapes of expected most typical trajectories. Figure 7a contains a clearly separated trajectories similar in shape to a dot and a worm. Figure 7b contains a circular shape in the center (larger than a typical dot) and there is a halo surrounding it, which affects CMOS sensor less than the core of the potential hit. Figure 7c looks like a typical worm, but the angle between its two parts is close to a right angle, which is unusual. Figure 7d also morphologically resembles a worm, however the trajectory forms a closed loop. Figure 7e contains a relatively wide rectilinear band, probably with a low energy deposit, which resembles a cloud. Figure 7f is probably the result of image file corruption because it looks like it consists of two images separated horizontally. Figure 7h contains a single circular area, but much larger than typical dot class representatives. In contrast, Figure 7g contains a large dot having an additional linear tail.

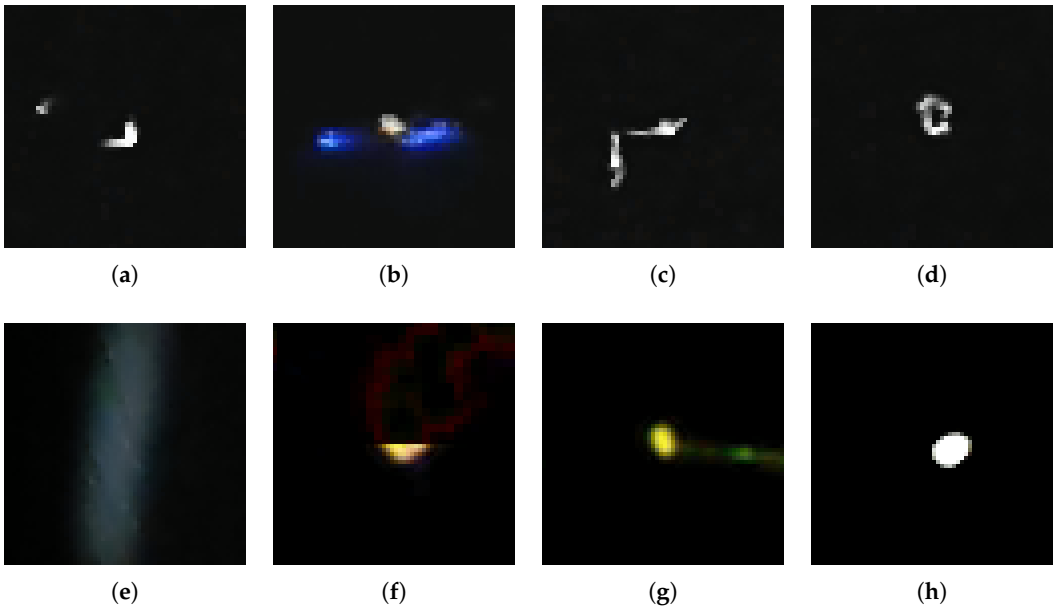

Figure 7. Examples of anomalies detected by the proposed method with parameters ($\alpha = 2.3$, $k = 5$), B. Replicate preprocessing, basic PCA. We chose them because they represent a variety of deviations from the typical shapes of expected most typical trajectories: (**a**) two separated signals, (**b**) energy deposit wit colored halo effect, (**c**) worm-like signal with unexpected right angle, (**d**) untypical closed loop trajectory, (**e**) wide band with low energy deposit, (**f**) probably corrupted image file, (**g**) colored energy deposit with tail, (**h**) dot-like signal with too large energy deposit.

5. Conclusions

In conclusion, the method proposed in this paper for detecting potential anomalous cosmic ray particle tracks in big data image dataset acquired by CMOS proved to be effective in terms of the returned results. The use of Incremental PCA allowed approximation of V matrix which might be updated at runtime. Incremental PCA results with almost identical embedding as basic PCA. This significantly reduces the necessary calculations in terms of memory complexity so that our method can be used for big data. The use of intuitive parameters of the potential anomalies detection algorithm based on object density in embedding space makes our method intuitive to use. By manipulating the pair (ϵ, k) in (4), we can explore outliers and calibrate the algorithm for our needs with polynomial computational complexity even if we do not use parallel computing. The proposed method (6) can also be used to find similar objects, which gives it the potential, for example, to be used in minimal distance-based classification and image database querying. This application is worth further investigation as it would allow interactive exploration of the whole CREDO experiment dataset in real time, which is an important issue in terms of science and cognition.

Author Contributions: Conceptualization: T.H. and M.P.; methodology: T.H.; software: T.H.; validation: T.H. and M.P.; formal analysis: T.H.; investigation, T.H. and M.P.; data curation: T.H.; writing—original draft preparation, T.H. and M.P.; writing—review and editing: T.H. and M.P.; visualization: T.H.; funding acquisition, T.H. and M.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Source codes can be downloaded from: https://github.com/browarsoftware/anomalies_bigdata accessed on 20 December 2023.

Acknowledgments: We would like to thank the CREDO Collaboration as a whole and the CREDO-ML research group in particular for providing a consistent subset of observational data for this publication.

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

References

1. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection: A survey. *ACM Comput. Surv. (CSUR)* **2009**, *41*, 1–58. [CrossRef]
2. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly Detection. In *Encyclopedia of Machine Learning and Data Mining*; Sammut, C., Webb, G.I., Eds.; Springer: Boston, MA, USA, 2016; pp. 1–15. [CrossRef]
3. Hodge, V.; Austin, J. A survey of outlier detection methodologies. *Artif. Intell. Rev.* **2004**, *22*, 85–126. [CrossRef]
4. Ben-Gal, I. Outlier detection. In *Data Mining and Knowledge Discovery Handbook*; Springer: Boston, MA, USA, 2005; pp. 131–146.
5. Ruff, L.; Kauffmann, J.R.; Vandermeulen, R.A.; Montavon, G.; Samek, W.; Kloft, M.; Dietterich, T.G.; Müller, K.R. A Unifying Review of Deep and Shallow Anomaly Detection. *Proc. IEEE* **2021**, *109*, 756–795. [CrossRef]
6. Pimentel, M.A.; Clifton, D.A.; Clifton, L.; Tarassenko, L. A review of novelty detection. *Signal Process.* **2014**, *99*, 215–249. [CrossRef]
7. Boukerche, A.; Zheng, L.; Alfandi, O. Outlier detection: Methods, models, and classification. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–37. [CrossRef]
8. Nassif, A.B.; Talib, M.A.; Nasir, Q.; Dakalbab, F.M. Machine Learning for Anomaly Detection: A Systematic Review. *IEEE Access* **2021**, *9*, 78658–78700. [CrossRef]
9. Pang, G.; Shen, C.; Cao, L.; Hengel, A.V.D. Deep Learning for Anomaly Detection: A Review. *ACM Comput. Surv.* **2021**, *54*. [CrossRef]
10. Goldman, A.; Cohen, I. Anomaly detection based on an iterative local statistics approach. *Signal Process.* **2004**, *84*, 1225–1229. [CrossRef]
11. Ahmed, T. Online anomaly detection using KDE. In Proceedings of the GLOBECOM 2009–2009 IEEE Global Telecommunications Conference, Honolulu, HI, USA, 30 November–4 December 2009; pp. 1–8.
12. Kim, J.; Scott, C.D. Robust kernel density estimation. *J. Mach. Learn. Res.* **2012**, *13*, 2529–2565.
13. Pang, G.; Aggarwal, C. Toward explainable deep anomaly detection. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Singapore, 14–18 August 2021; pp. 4056–4057.
14. Yuan, S.; Wu, X. Trustworthy anomaly detection: A survey. *arXiv* **2022**, arXiv:2202.07787.
15. Kwon, D.; Kim, H.; Kim, J.; Suh, S.C.; Kim, I.; Kim, K.J. A survey of deep learning-based network anomaly detection. *Clust. Comput.* **2019**, *22*, 949–961. [CrossRef]
16. Fernandes, G.; Rodrigues, J.J.; Carvalho, L.F.; Al-Muhtadi, J.F.; Proença, M.L. A comprehensive survey on network anomaly detection. *Telecommun. Syst.* **2019**, *70*, 447–489. [CrossRef]
17. Hwang, R.H.; Peng, M.C.; Huang, C.W.; Lin, P.C.; Nguyen, V.L. An unsupervised deep learning model for early network traffic anomaly detection. *IEEE Access* **2020**, *8*, 30387–30399. [CrossRef]
18. Burgueño, J.; de-la Bandera, I.; Mendoza, J.; Palacios, D.; Morillas, C.; Barco, R. Online anomaly detection system for mobile networks. *Sensors* **2020**, *20*, 7232. [CrossRef] [PubMed]
19. Fotiadou, K.; Velivassaki, T.H.; Voulkidis, A.; Skias, D.; Tsekeridou, S.; Zahariadis, T. Network traffic anomaly detection via deep learning. *Information* **2021**, *12*, 215. [CrossRef]
20. Joudaki, H.; Rashidian, A.; Minaei-Bidgoli, B.; Mahmoodi, M.; Geraili, B.; Nasiri, M.; Arab, M. Using data mining to detect health care fraud and abuse: A review of literature. *Glob. J. Health Sci.* **2015**, *7*, 194. [CrossRef]
21. Ahmed, M.; Mahmood, A.N.; Hu, J. A survey of network anomaly detection techniques. *J. Netw. Comput. Appl.* **2016**, *60*, 19–31. [CrossRef]
22. Zheng, Y.J.; Zhou, X.H.; Sheng, W.G.; Xue, Y.; Chen, S.Y. Generative adversarial network based telecom fraud detection at the receiving bank. *Neural Netw.* **2018**, *102*, 78–86. [CrossRef] [PubMed]
23. Jiang, J.; Chen, J.; Gu, T.; Choo, K.K.R.; Liu, C.; Yu, M.; Huang, W.; Mohapatra, P. Anomaly detection with graph convolutional networks for insider threat and fraud detection. In Proceedings of the MILCOM 2019–2019 IEEE Military Communications Conference (MILCOM), Norfolk, VA, USA, 12–14 November 2019; pp. 109–114.

24. Pourhabibi, T.; Ong, K.L.; Kam, B.H.; Boo, Y.L. Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decis. Support Syst.* **2020**, *133*, 113303. [CrossRef]
25. Hilal, W.; Gadsden, S.A.; Yawney, J. Financial fraud: A review of anomaly detection techniques and recent advances. *Expert Syst. Appl.* **2022**, *193*, 116429. [CrossRef]
26. Favarelli, E.; Giorgetti, A. Machine learning for automatic processing of modal analysis in damage detection of bridges. *IEEE Trans. Instrum. Meas.* **2020**, *70*, 1–13. [CrossRef]
27. Chow, J.K.; Su, Z.; Wu, J.; Tan, P.S.; Mao, X.; Wang, Y.H. Anomaly detection of defects on concrete structures with the convolutional autoencoder. *Adv. Eng. Inform.* **2020**, *45*, 101105. [CrossRef]
28. Hong, G.; Suh, D. Supervised-learning-based intelligent fault diagnosis for mechanical equipment. *IEEE Access* **2021**, *9*, 116147–116162. [CrossRef]
29. Fourlas, G.K.; Karras, G.C. A survey on fault diagnosis and fault-tolerant control methods for unmanned aerial vehicles. *Machines* **2021**, *9*, 197. [CrossRef]
30. Schmidt, S.; Gryllias, K.C. The anomalous and smoothed anomalous envelope spectra for rotating machine fault diagnosis. *Mech. Syst. Signal Process.* **2021**, *158*, 107770. [CrossRef]
31. Velasco-Gallego, C.; Lazakis, I. RADIS: A real-time anomaly detection intelligent system for fault diagnosis of marine machinery. *Expert Syst. Appl.* **2022**, *204*, 117634. [CrossRef]
32. Adams, D.M.; Ricci, K.W. Vascular anomalies: Diagnosis of complicated anomalies and new medical treatment options. *Hematol. Clin.* **2019**, *33*, 455–470.
33. Anyamba, A.; Chretien, J.P.; Britch, S.C.; Soebiyanto, R.P.; Small, J.L.; Jepsen, R.; Forshey, B.M.; Sanchez, J.L.; Smith, R.D.; Harris, R.; et al. Global disease outbreaks associated with the 2015–2016 El Niño event. *Sci. Rep.* **2019**, *9*, 1930. [CrossRef]
34. Ouyang, X.; Karanam, S.; Wu, Z.; Chen, T.; Huo, J.; Zhou, X.S.; Wang, Q.; Cheng, J.Z. Learning hierarchical attention for weakly-supervised chest X-ray abnormality localization and diagnosis. *IEEE Trans. Med. Imaging* **2020**, *40*, 2698–2710. [CrossRef]
35. Fernando, T.; Gammulle, H.; Denman, S.; Sridharan, S.; Fookes, C. Deep learning for medical anomaly detection–a survey. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–37. [CrossRef]
36. Han, C.; Rundo, L.; Murao, K.; Noguchi, T.; Shimahara, Y.; Milacski, Z.Á.; Koshino, S.; Sala, E.; Nakayama, H.; Satoh, S. MADGAN: Unsupervised medical anomaly detection GAN using multiple adjacent brain MRI slice reconstruction. *BMC Bioinform.* **2021**, *22*, 31. [CrossRef] [PubMed]
37. Bondur, V.; Mokhov, I.; Voronova, O.; Sitnov, S. Satellite monitoring of Siberian wildfires and their effects: Features of 2019 anomalies and trends of 20-year changes. *Dokl. Earth Sci.* **2020**, *492*, 370–375. [CrossRef]
38. Peterson, K.T.; Sagan, V.; Sloan, J.J. Deep learning-based water quality estimation and anomaly detection using Landsat-8/Sentinel-2 virtual constellation and cloud computing. *GISci. Remote Sens.* **2020**, *57*, 510–525. [CrossRef]
39. Tang, M.; Ji, W.Q.; Chu, X.; Wu, A.; Chen, C. Reconstructing crustal thickness evolution from europium anomalies in detrital zircons. *Geology* **2021**, *49*, 76–80. [CrossRef]
40. Camps-Valls, G.; Tuia, D.; Zhu, X.X.; Reichstein, M. *Deep Learning for the Earth Sciences: A Comprehensive Approach to Remote Sensing, Climate Science and Geosciences*; John Wiley & Sons: Hoboken, NJ, USA, 2021.
41. Xu, J.; Zheng, Y.; Mao, Y.; Wang, R.; Zheng, W.S. Anomaly detection on electroencephalography with self-supervised learning. In Proceedings of the 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Seoul, Republic of Korea, 16–19 December 2020; pp. 363–368.
42. Liu, Y.; Chen, Y.; Han, L. Bioinformatics: Advancing biomedical discovery and innovation in the era of big data and artificial intelligence. *Innov. Med.* **2023**, *1*, 100012. [CrossRef]
43. Mandal, A.K.; Sarma, P.K.D.; Dehuri, S. A Study of Bio-inspired Computing in Bioinformatics: A State-of-the-art Literature Survey. *Open Bioinform. J.* **2023**, *16*, e187503622305100.
44. Ohkura, N.; Sakaguchi, S. Transcriptional and epigenetic basis of Treg cell development and function: Its genetic anomalies or variations in autoimmune diseases. *Cell Res.* **2020**, *30*, 465–474. [CrossRef]
45. Bedei, I.; Wolter, A.; Weber, A.; Signore, F.; Axt-Fliedner, R. Chances and challenges of new genetic screening technologies (NIPT) in prenatal medicine from a clinical perspective: A narrative review. *Genes* **2021**, *12*, 501. [CrossRef]
46. Nachman, B.; Shih, D. Anomaly detection with density estimation. *Phys. Rev. D* **2020**, *101*, 075042. [CrossRef]
47. Andreassen, A.; Nachman, B.; Shih, D. Simulation assisted likelihood-free anomaly detection. *Phys. Rev. D* **2020**, *101*, 095004. [CrossRef]
48. Finke, T.; Krämer, M.; Morandini, A.; Mück, A.; Oleksiyuk, I. Autoencoders for unsupervised anomaly detection in high energy physics. *J. High Energy Phys.* **2021**, *2021*, 161. [CrossRef]
49. Atkinson, O.; Bhardwaj, A.; Englert, C.; Ngairangbam, V.S.; Spannowsky, M. Anomaly detection with convolutional graph neural networks. *J. High Energy Phys.* **2021**, *2021*, 80. [CrossRef]
50. Mikuni, V.; Nachman, B.; Shih, D. Online-compatible unsupervised nonresonant anomaly detection. *Phys. Rev. D* **2022**, *105*, 055006. [CrossRef]
51. Reyes, E.; Estévez, P.A. Transformation based deep anomaly detection in astronomical images. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–8.
52. Lochner, M.; Bassett, B.A. ASTRONOMALY: Personalised active anomaly detection in astronomical data. *Astron. Comput.* **2021**, *36*, 100481. [CrossRef]

53. Dere, S.; Fatima, M.; Jagtap, R.; Inamdar, U.; Shardoor, N.B. Anomaly Detection in Astronomical Objects of Galaxies Using Deep Learning. In Proceedings of the 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 19–20 March 2021; Volume 1, pp. 702–706.
54. Villar, V.A.; Cranmer, M.; Berger, E.; Contardo, G.; Ho, S.; Hosseinzadeh, G.; Lin, J.Y.Y. A deep-learning approach for live anomaly detection of extragalactic transients. *Astrophys. J. Suppl. Ser.* **2021**, *255*, 24. [CrossRef]
55. Mandrikova, O.; Mandrikova, B. Hybrid Method for Detecting Anomalies in Cosmic ray Variations Using Neural Networks Autoencoder. *Symmetry* **2022**, *14*, 744. [CrossRef]
56. Mesarcik, M.; Boonstra, A.J.; Iacobelli, M.; Ranguelova, E.; de Laat, C.; van Nieuwpoort, R. The ROAD to discovery: Machine-learning-driven anomaly detection in radio astronomy spectrograms. *Astron. Astrophys.* **2023**, *680*, A74. [CrossRef]
57. Fraser, K.; Homiller, S.; Mishra, R.K.; Ostdiek, B.; Schwartz, M.D. Challenges for unsupervised anomaly detection in particle physics. *J. High Energy Phys.* **2022**, *2022*, 66. [CrossRef]
58. Kuusela, M.; Vatanen, T.; Malmi, E.; Raiko, T.; Aaltonen, T.; Nagai, Y. Semi-supervised anomaly detection–towards model-independent searches of new physics. *J. Phys. Conf. Ser.* **2012**, *368*, 012032. [CrossRef]
59. Stein, G.; Seljak, U.; Dai, B. Unsupervised in-distribution anomaly detection of new physics through conditional density estimation. *arXiv* **2020**, arXiv:2012.11638.
60. Crispim Romão, M.; Castro, N.F.; Pedro, R. Finding new physics without learning about it: Anomaly detection as a tool for searches at colliders. *Eur. Phys. J. C* **2021**, *81*, 27. [CrossRef]
61. Poy, A.B.; Boterenbrood, H.; Burckhart, H.; Cook, J.; Filimonov, V.; Franz, S.; Gutzwiller, O.; Hallgren, B.; Khomutnikov, V.; Schlenker, S.; et al. The detector control system of the ATLAS experiment. *J. Instrum.* **2008**, *3*, P05006. [CrossRef]
62. Adolphi, R. The CMS experiment at the CERN LHC. *Jinst* **2008**, *803*, S08004.
63. Kalweit, A.; The ALICE Collaboration. Particle identification in the ALICE experiment. *J. Phys. G Nucl. Part. Phys.* **2011**, *38*, 124073. [CrossRef]
64. Brust, C.; Katz, A.; Lawrence, S.; Sundrum, R. SUSY, the Third Generation and the LHC. *J. High Energy Phys.* **2012**, *2012*, 103. [CrossRef]
65. Hemsing, E.; Marcus, G.; Fawley, W.; Schoenlein, R.; Coffee, R.; Dakovski, G.; Hastings, J.; Huang, Z.; Ratner, D.; Raubenheimer, T.; et al. Soft X-ray seeding studies for the SLAC Linac Coherent Light Source II. *Phys. Rev. Accel. Beams* **2019**, *22*, 110701. [CrossRef]
66. Grames, J.; Higinbotham, D.W.; Montgomery, H.E. Thomas Jefferson National Accelerator Facility. *Nucl. Phys. News* **2010**, *20*, 6–13. [CrossRef]
67. Li, W. Heavy Gas Cherenkov Construction for Hall C at Thomas Jefferson National Accelerator Facility. *arXiv* **2023**, arXiv:2304.10016.
68. Hasegawa, K.; Hayashi, N.; Oguri, H.; Yamamoto, K.; Kinsho, M.; Yamazaki, Y.; Naito, F.; Koseki, T.; Yamamoto, N.; Yoshii, M. Performance and Status of the J-PARC Accelerators. In Proceedings of the 8th International Particle Accelerator Conference, Copenhagen, Denmark, 14–19 May 2017.
69. Hachiya, T. J-PARC heavy ion experiment. *Int. J. Mod. Phys. E* **2020**, *29*, 2040005. [CrossRef]
70. Kampert, K.H.; Alejandro Mostafa, M.; Zas, E.; Pierre Auger Collaboration. Multi-messenger physics with the Pierre Auger Observatory. *Front. Astron. Space Sci.* **2019**, *6*, 24. [CrossRef]
71. Verzi, V.; Pierre Auger Collaboration. Measurement of the energy spectrum of ultra-high energy cosmic rays using the Pierre Auger Observatory. In Proceedings of the 36th International Cosmic Ray Conference, Madison, WI, USA, 24 July–1 August 2019; SISSA Medialab: Trieste TS, Italy, 2021; Volume 358, p. 450.
72. Aartsen, M.G.; Ackermann, M.; Adams, J.; Aguilar, J.; Ahlers, M.; Ahrens, M.; Altmann, D.; Andeen, K.; Anderson, T.; Ansseau, I.; et al. The IceCube Neutrino Observatory: Instrumentation and online systems. *J. Instrum.* **2017**, *12*, P03012. [CrossRef]
73. Tokuno, H.; Tameda, Y.; Takeda, M.; Kadota, K.; Ikeda, D.; Chikawa, M.; Fujii, T.; Fukushima, M.; Honda, K.; Inoue, N.; et al. New air fluorescence detectors employed in the Telescope Array experiment. *Nucl. Instrum. Methods Phys. Res. Sect. A Accel. Spectrom. Detect. Assoc. Equip.* **2012**, *676*, 54–65. [CrossRef]
74. Bibrzycki, L.; Burakowski, D.; Homola, P.; Piekarczyk, M.; Niedźwiecki, M.; Rzecki, K.; Stuglik, S.; Tursunov, A.; Hnatyk, B.; Castillo, D.E.A.; et al. Towards A Global Cosmic Ray Sensor Network: CREDO Detector as the First Open-Source Mobile Application Enabling Detection of Penetrating Radiation. *Symmetry* **2020**, *12*, 1802. [CrossRef]
75. Kumar, R. Tracking Cosmic Rays by CRAYFIS (Cosmic Rays Found in Smartphones) Global Detector. In Proceedings of the 34th International Cosmic Ray Conference (ICRC2015), The Hague, The Netherlands, 30 July–6 August 2015; Volume 34, p. 1234.
76. Whiteson, D.; Mulhearn, M.; Shimmin, C.; Cranmer, K.; Brodie, K.; Burns, D. Searching for ultra-high energy cosmic rays with smartphones. *Astropart. Phys.* **2016**, *79*, 1–9. [CrossRef]
77. Vandenbroucke, J.; Bravo, S.; Karn, P.; Meehan, M.; Plewa, M.; Ruggles, T.; Schultz, D.; Peacock, J.; Simons, A.L. Detecting particles with cell phones: The Distributed Electronic Cosmic-ray Observatory. *arXiv* **2015**, arXiv:1510.07665.
78. Vandenbroucke, J.; BenZvi, S.; Bravo, S.; Jensen, K.; Karn, P.; Meehan, M.; Peacock, J.; Plewa, M.; Ruggles, T.; Santander, M.; et al. Measurement of cosmic-ray muons with the Distributed Electronic Cosmic-ray Observatory, a network of smartphones. *J. Instrum.* **2016**, *11*, P04019. [CrossRef]
79. Homola, P.; Beznosko, D.; Bhatta, G.; Bibrzycki, L.; Borczyńska, M.; Bratek, L.; Budnev, N.; Burakowski, D.; Alvarez-Castillo, D.E.; Almeida Cheminant, K.; et al. Cosmic-Ray Extremely Distributed Observatory. *Symmetry* **2020**, *12*, 1835. [CrossRef]

80. Karbowiak, M.; Wibig, T.; Alvarez Castillo, D.; Beznosko, D.; Duffy, A.R.; Góra, D.; Homola, P.; Kasztelan, M.; Niedźwiecki, M. Determination of zenith angle dependence of incoherent cosmic ray muon flux using smartphones of the CREDO Project. *Appl. Sci.* **2021**, *11*, 1185. [CrossRef]
81. Albin, E.K.; Whiteson, D. Feasibility of Correlated Extensive Air Shower Detection with a Distributed Cosmic-Ray Network. *Astrophys. J.* **2023**, *954*, 106. [CrossRef]
82. Winter, M.; Bourbeau, J.; Bravo, S.; Campos, F.; Meehan, M.; Peacock, J.; Ruggles, T.; Schneider, C.; Simons, A.L.; Vandenbroucke, J. Particle identification in camera image sensors using computer vision. *Astropart. Phys.* **2019**, *104*, 42–53. [CrossRef]
83. Karbowiak, M.; Wibig, T.; Alvarez-Castillo, D.; Beznosko, D.; Duffy, A.R.; Góra, D.; Homola, P.; Kasztelan, M.; Niedźwiecki, M. The first CREDO registration of extensive air shower. *Phys. Educ.* **2020**, *55*, 055021. [CrossRef]
84. Karbowiak, M.; Orzechowski, M.; Wibig, T.; Bibrzycki, Ł.; Kovacs, P.; Piekarczyk, M.; Stasielak, J.; Stuglik, S.; Sushchov, O. Small shower array for education purposes-the CREDO-Maze Project. *Proc. Sci.* **2021**, *395*, 199.
85. Credo; Pryga, J.S.; Wozniak, K.W.; Bibrzycki, L.; Homola, P.; Niedźwiedzki, J.; Alvarez-Castillo, D.; Hachaj, T.; Hnatyk, B.; Piekarczyk, M.; et al. Detection of Extensive Air Showers with small array–measurement and estimations. In Proceedings of the 38th International Cosmic Ray Conference (ICRC2023), Nagoya, Japan, 26 July–3 August 2023; p. 382. [CrossRef]
86. Wibig, T.; Karbowiak, M. CREDO-Maze Cosmic Ray Mini-Array for Educational Purposes. *Symmetry* **2022**, *14*, 500. [CrossRef]
87. Lawie, M.R.; Vosper, F.; Cremonesi, L.; Booth, A. Exploring the Sensitivity of MiniPix Devices to the Detection of a Variety of Particles. *Emerg. Minds J. Stud. Res.* **2023**, *1*, 90–100. [CrossRef]
88. Kokate, U.; Deshpande, A.; Mahalle, P.; Patil, P. Data stream clustering techniques, applications, and models: comparative analysis and discussion. *Big Data Cogn. Comput.* **2018**, *2*, 32. [CrossRef]
89. Bahri, M.; Bifet, A.; Gama, J.; Gomes, H.M.; Maniu, S. Data stream analysis: Foundations, major tasks and tools. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2021**, *11*, e1405. [CrossRef]
90. Hachaj, T.; Piekarczyk, M. The Practice of Detecting Potential Cosmic Rays Using CMOS Cameras: Hardware and Algorithms. *Sensors* **2023**, *23*, 4858. [CrossRef]
91. Piekarczyk, M.; Bar, O.; Bibrzycki, L.; Niedźwiecki, M.; Rzecki, K.; Stuglik, S.; Andersen, T.; Budnev, N.M.; Alvarez-Castillo, D.E.; Cheminant, K.A.; et al. CNN-Based Classifier as an Offline Trigger for the CREDO Experiment. *Sensors* **2021**, *21*, 4804. [CrossRef]
92. Bibrzycki, Ł.; Bibrzycki, L.; Alvarez-Castillo, D.; Bar, O.; Gora, D.; Homola, P.; Kovacs, P.; Niedźwiecki, M.; Piekarczyk, M.; Rzecki, K.; et al. Machine learning aided noise filtration and signal classification for CREDO experiment. In Proceedings of the 37th International Cosmic Ray Conference (ICRC2021), Berlin, Germany, 12–23 July 2021; p. 227. [CrossRef]
93. Pabian, M.; Rzepka, D.; Bibrzycki, L.; Pawlak, M. Differentiating signal from artefacts in cosmic ray detection: Applying Siamese spiking neural networks to CREDO experimental data. *Measurement* **2023**, *220*, 113273. [CrossRef]
94. Hachaj, T.; Bibrzycki, L.; Piekarczyk, M. Recognition of Cosmic Ray Images Obtained from CMOS Sensors Used in Mobile Phones by Approximation of Uncertain Class Assignment with Deep Convolutional Neural Network. *Sensors* **2021**, *21*, 1963. [CrossRef]
95. Hachaj, T.; Piekarczyk, M.; Bibrzycki, Ł. Deep Neural Network Architecture for Low-Dimensional Embedding and Classification of Cosmic Ray Images Obtained from CMOS Cameras. In *Proceedings of the Neural Information Processing*; Mantoro, T., Lee, M., Ayu, M.A., Wong, K.W., Hidayanto, A.N., Eds.; Springer: Cham, Switzerland, 2021; pp. 307–316.
96. Bar, O.; Bibrzycki, L.; Niedźwiecki, M.; Piekarczyk, M.; Rzecki, K.; Sośnicki, T.; Stuglik, S.; Frontczak, M.; Homola, P.; Alvarez-Castillo, D.E.; et al. Zernike Moment Based Classification of Cosmic Ray Candidate Hits from CMOS Sensors. *Sensors* **2021**, *21*, 7718. [CrossRef]
97. Hachaj, T.; Piekarczyk, M.; Wąs, J. Searching of Potentially Anomalous Signals in Cosmic-Ray Particle Tracks Images Using Rough k-Means Clustering Combined with Eigendecomposition-Derived Embedding. In *Proceedings of the Rough Sets*; Campagner, A., Urs Lenz, O., Xia, S., Ślęzak, D., Wąs, J.,Yao, J., Eds.; Springer: Cham, Switzerland, 2023; pp. 431–445.
98. 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]
99. Gewers, F.L.; Ferreira, G.R.; Arruda, H.F.D.; Silva, F.N.; Comin, C.H.; Amancio, D.R.; Costa, L.d.F. Principal component analysis: A natural approach to data exploration. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–34. [CrossRef]
100. Greenacre, M.; Groenen, P.J.; Hastie, T.; d'Enza, A.I.; Markos, A.; Tuzhilina, E. Principal component analysis. *Nat. Rev. Methods Prim.* **2022**, *2*, 100. [CrossRef]
101. Ross, D.A.; Lim, J.; Lin, R.S.; Yang, M.H. Incremental learning for robust visual tracking. *Int. J. Comput. Vis.* **2008**, *77*, 125–141. [CrossRef]
102. Dubey, D.; Tomar, G.S. Image alignment in pose variations of human faces by using corner detection method and its application for PIFR system. *Wirel. Pers. Commun.* **2022**, *124*, 147–162. [CrossRef]
103. Gogić, I.; Ahlberg, J.; Pandžić, I.S. Regression-based methods for face alignment: A survey. *Signal Process.* **2021**, *178*, 107755. [CrossRef]
104. Chaudhary, U.N.; Kelly, C.N.; Wesorick, B.R.; Reese, C.M.; Gall, K.; Adams, S.B.; Sapiro, G.; Di Martino, J.M. Computational and image processing methods for analysis and automation of anatomical alignment and joint spacing in reconstructive surgery. *Int. J. Comput. Assist. Radiol. Surg.* **2022**, *17*, 541–551. [CrossRef]
105. Hachaj, T.; Mazurek, P. Comparative Analysis of Supervised and Unsupervised Approaches Applied to Large-Scale "In The Wild" Face Verification. *Symmetry* **2020**, *12*, 1832. [CrossRef]
106. Bradski, G. The openCV library. *Dr. Dobb's J. Softw. Tools Prof. Program.* **2000**, *25*, 120–123.

107. Wang, J.; Chang, C.I. Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis. *IEEE Trans. Geosci. Remote Sens.* **2006**, *44*, 1586–1600. [CrossRef]
108. Zhang, P.; Li, C.; Wang, C. VisCode: Embedding Information in Visualization Images using Encoder-Decoder Network. *IEEE Trans. Vis. Comput. Graph.* **2021**, *27*, 326–336. [CrossRef]
109. Saravanan, C. Color image to grayscale image conversion. In Proceedings of the 2010 Second International Conference on Computer Engineering and Applications, Bali, Indonesia, 19–21 March 2010; Volume 2, pp. 196–199.
110. Hachaj, T.; Koptyra, K.; Ogiela, M.R. Eigenfaces-Based Steganography. *Entropy* **2021**, *23*, 273. [CrossRef]
111. Ward, J.H. Hierarchical Grouping to Optimize an Objective Function. *J. Am. Stat. Assoc.* **1963**, *58*, 236–244. [CrossRef]
112. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, OR, USA, 2–4 August 1996; Volume 96, pp. 226–231.
113. Levey, A.; Lindenbaum, M. Sequential Karhunen-Loeve basis extraction and its application to images. *IEEE Trans. Image Process.* **2000**, *9*, 1371–1374. [CrossRef] [PubMed]
114. Chan, T.F.; Golub, G.H.; LeVeque, R.J. Algorithms for computing the sample variance: Analysis and recommendations. *Am. Stat.* **1983**, *37*, 242–247. [CrossRef]
115. Verma, V.; Aggarwal, R.K. A comparative analysis of similarity measures akin to the Jaccard index in collaborative recommendations: Empirical and theoretical perspective. *Soc. Netw. Anal. Min.* **2020**, *10*, 43. [CrossRef]
116. Mizuno, S.; Yamaguchi, T.; Fukushima, A.; Matsuyama, Y.; Ohashi, Y. Overlap coefficient for assessing the similarity of pharmacokinetic data between ethnically different populations. *Clin. Trials* **2005**, *2*, 174–181. [CrossRef] [PubMed]