Intelligent analysis of the list of technological operations for the problem of developing of functional simulators stowage and installation-mooring of the landing gear

L A Demidova ¹ and B Yu Mordakin ²

¹ MIREA – Russian Technological University, 78, Vernadsky avenye, Moscow, 119454, Russia
² Ryazan Higher Airborne Command School named after Army General V. F. Margelov, 1 Army General V.F. Margelov sq., Ryazan, 390031, Russia

E-mail: mbour@mail.ru

Abstract. The aim of the article is to investigate the number and assortment of training solutions required to perform technical operations involved in parachute packing and landing platform mounting and mooring. The article proposes a novel approach to the identification of the number and value of the attributes of technical operations, which takes into consideration objective information and expert assessment results. The article presents an algorithm aimed at the identification of the number and assortment of training solutions based on the algorithm of DBSCAN cluster analysis and promoting the calibration of expert assessments in order to improve clustering quality. The results of experimental research prove the feasibility of dividing technical operations into clusters whose quantity and composition are predicted by experts and are practically relevant.

1. Introduction
The technological procedure of parachute packing and landing platform mounting and mooring involves a multitude of separate technical operations such as: check, defects identification, detachment, attachment, untying, tying, fastening, adjustment, stretching, etc. Technical operations differ in their function, complexity, number of repetitions, tools required to perform them, etc. In addition to standard tools (hammer, screwdriver, and flat pliers), one needs specialized tools such as pull-up cords, ramming tools, wire loops, rigging screws.

It is obvious that in order to ensure the acquisition of practical knowledge required for parachute packing and landing platform mounting and mooring, it is essential to develop training solutions meant to provide practice in coping with technical operations.

The issue of developing training solutions involving different functionality has been investigated by a number of researchers. Thus, in [1-4] they discuss various approaches to the identification of characteristics required for motion cueing mechanisms generating translational and angular motions as the
Stewart 6-3 (SPM) platform mechanism. These motions ensure the approximation of specific force and angular velocity characteristic of a pilot’s flight simulation experience.

The researchers compare flight characteristics of a flight simulator and an aircraft. If simulators are to be used for typical and unusual scenarios, it is essential to ensure a high degree of accordance between simulated and real flight both on the ground and in the air. The authors discuss a full-scale systemic approach to technical operations identification and present a classification of flight modes and ground contact for the whole assortment of technical operations from aircraft deceleration to taxiing, take off, landing, and parking.

The authors assess training efficiency and cost-effectiveness of visual flight simulation systems for military aircraft and other training solutions.

2. Compiling lists of technical operations and identifying the value of their attributes

The analysis of the technological procedure of parachute packing and landing platform mounting and mooring enables the authors to single out 245 separate technical operations (aka basic steps). Having mastered them, one can ensure successful completion of the overall technological procedure of preparing cargo and equipment for airdropping. It is obvious that to ensure the acquisition of practical knowledge required for the performance of technical operations, it is essential to develop a certain number of training solutions (from 4 to 8). Training simulators ensure specialists’ universal training. Therefore, it is crucial that technical operations should be grouped in a certain order with due consideration to their similarities singled out and substantiated by experts specializing in the preparation of cargo and equipment for airdropping.

To profoundly investigate the technological procedure of parachute packing and landing platform mounting and mooring, one should analyze a list of attributes that are expected to characterise each technical operation. It is also essential to create an exhaustive list of attributes unique to each operation. It will enable one to group similar technical operations into clusters characterized by certain individuality and, therefore, different from other clusters.

The procedure of attribute investigation and classification is hard to formalize and is usually performed heuristically. The assortment of attributes is dependent on the classification results, i.e. the process of shaping a set of characteristic features can be iterative.

The analysis of 215 technical operations (basic steps performed by specialists) enables the authors to single out five attributes characterizing technical operations.

1. Attribute characterizing the type of operation: check, defects identification, mounting, stowage, control, regulation, dismounting.
2. Attribute characterizing the complexity of technological equipment (tools) required to perform a technical operation.
3. Attribute characterizing the time of operation and its complexity.
4. Attribute characterizing the number of iterations required to complete a technological procedure.
5. Attribute characterizing the required level of specialists’ proficiency.

As a result, every $i$-th technical operation can be described by the $X_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5})$ vector, where each $j$-th element corresponds to an attribute listed above ($i = 1,245; j = 1,5$). Therefore, $x_{ij}$ represents quantitative assessment of the $j$-th attribute of the $i$-th operation, the $X_i \in R^5$ vector reflects individual peculiarities of the $i$-th operation, while the combination of $X_i$ ($i = 1,245$) vectors in the $R^5$ space describes the technological procedure of parachute packing and landing platform mounting and mooring.

Each attribute can be associated with a certain quantitative indicator.
The third and the forth attributes are logically associated with the time required to perform technical operations and the number of iterations required to complete a technological procedure, respectively. To identify the quantitative indicators of the other three attributes (the first, the second, and the fifth) in accordance with expert assessments, it is recommended to employ assessment scales.

To identify the quantitative indicator of the first attribute (the attribute characterizing the type of technical operation) according to a 9-point assessment scale, we will assign 1 point to the operation of check; 2 points to the operation of defect identification; 3 points to the operation of mounting, 4 points to the operation of tool and equipment preparation; 5 point to the operation of stowage, 6 points to the operation of preliminary assessment of positioning; 7 points to operations of fastening, fixation, locking and strapping; 8 points to regulatory operations; 9 points to mounting and mooring. As a result, the quantitative code of an operation will be an integer value within the specified range (1-9).

When identifying the quantitative indicator of the second attribute [the attribute of the complexity of technological equipment (tool) required for the performance of a technical operation] we will assign 2 points if a technical operation can be performed using one tool (packing paddle, packing hook, molar strap, eye splice, knife, chuck, key, screw driver, hammer, etc.); we will assign 4 points if two or four tools must be used; we will assign 6 points if the performance of an operation requires complex equipment (clamp, molar strap, bodkin, etc.).

When identifying the quantitative indicator for the fifth attribute (the attribute characterizing the required level of specialists’ proficiency), we will assign 2 points if an operation can be performed by a low-skilled worker, 5 points if an operation must be performed by a middle-skill worker, 8 points if an operation can only be performed by a high-skilled worker.

Table 1. The values of the attributes of technical operations of parachute packing and landing platform mounting and mooring.

| No | Action                                | Signs of | | | |
|----|---------------------------------------|----------|--|--|--|
| 1  | Fold the right half canopy cells      | $x_{1,1}$ | $x_{1,2}$ | $x_{1,3}$ | $x_{1,4}$ | $x_{1,5}$ |
| 2  | Fold the left half canopy cells       | $x_{2,1}$ | $x_{2,2}$ | $x_{2,3}$ | $x_{2,4}$ | $x_{2,5}$ |
| ... | ........................................ | ... | ... | ... | ... | ... |
| 215| Check the position of the cutaway handle | $x_{245,1}$ | $x_{245,1}$ | $x_{245,1}$ | $x_{245,1}$ | $x_{245,1}$ |

One can compile a table including all numeric values of attributes for each technical operation (table 1) and use it to collect data that can further be used to divide technical operations into clusters to further develop training simulators aimed at training one to perform typical technical operations that are similar in their function (performance).

To classify data describing technical operations, it is advisable to use an algorithm of clusterisation [5 – 10]. The procedure may involve a number of difficulties associated with the selection of an algorithm (from a wide variety of clusterisation algorithms) that best divides the analyzed data into clusters. An adequate solution to the task of choosing an algorithm of clusterisation might be facilitated by a quality data visualization, which is, however, impossible, for the analyzed data are high-dimensional. Nevertheless, the problem of visualisation can be solved if we use the so called algorithms of nonlinear dimensionality reduction, which presupposes the transformation of data from a high-dimensional space into a low-dimensional space (for instance, a two-dimensional space).
3. An approach to technical operations data visualisation

3.1. Nonlinear dimensionality reduction on the basis of the t-SNE-algorithm

Nowadays, to solve the task of data dimensionality reduction, researchers actively use various algorithms of nonlinear dimensionality reduction, which enable researchers to present multidimensional data in a two-dimensional or three-dimensional metric space without violating the proportionality of distance between points and the structure of clusters [11 – 18]. It should be noted that one can vary parameter values of these algorithms ensuring that both local and global characteristics are taken into consideration. Python provides solutions that can be used to deal with many of these algorithms. It is expedient to mention such algorithms as the t-SNE-algorithm [17] and the UMAP-algorithm [18], which have recently gained great popularity.

The authors of the research employed the t-SNE-algorithm (t-distributed Stochastic Neighbor Embedding) [17], which is an algorithm of machine learning which enables nonlinear dimensionality reduction and is designed to visualize data of high dimensionality by transforming them from a high-dimensional space into a low-dimensional space. The t-SNE-algorithm allows representing an object of a high-dimensional space as an object of a two-dimensional or three-dimensional space in such a way so that similar objects are represented as closely located points while dissimilar objects are represented as distantly located points.

If \(x_1, \ldots, x_n\) are points in a high-dimensional space which should be transformed into a low-dimensional space, let us compare them to points \(y_1, \ldots, y_n\).

The t-SNE-algorithm is based on the SNE-algorithm [17].

The SNE-algorithm ensures that if we know the distance between the \(x_i\) point and the \(x_j\) point in a metrical space of high dimensionality, we can calculate the \(p_{ij}\) conditional probability, i.e. the probability of the \(x_i\) point to neighbor the \(x_j\) point rather than any other point, provided that neighbors are chosen on the basis of the Gaussian probability distribution with the mathematical expectation of \(x_i\).

The \(p_{ij}\) conditional probability of the \(x_i\) point to neighbour the \(x_j\) point decreases with the increase of the distance between them and can be calculated as [17]:

\[
p_{ij} = \frac{\exp\left(-\frac{\|x_i - x_j\|^2}{2 \cdot \sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{\|x_i - x_k\|^2}{2 \cdot \sigma_i^2}\right)} \quad (i \neq j),
\]

where \(x_i, x_j, x_k\) are points in the original high-dimensional Euclidian space; \(\sigma_i\) is variance (standard deviation) of the Gaussian probability distribution with the centre at \(x_i\). For all \(i\): \(p_{ii} = 0\) and \(\sum_j p_{ij} = 1\).

The SNE-algorithm shows a similar distribution of points in a metrical space of low dimensionality, into which points of a high-dimensional space are placed.
\[ q_{ji} = \frac{\exp\left(-\|y_j - y_i\|^2\right)}{\sum_{k \neq i} \exp(-\|y_j - y_k\|^2)}, \]  

where \( y_i, y_j, y_k \) are points in a low-dimensional Euclidean space; \( q_{ji} = 0, \sum_j q_{ji} = 1 \)

It is assumed that it is feasible to use the Gaussian probability distribution, however, it is believed that \( \sigma_i^2 = \frac{1}{2} (i = 1, \ldots, n) \).

If low-dimensional points \( y_i \) and \( y_j \) correctly emulate the similarity between the original high-dimensional points \( x_i \) and \( x_j \), the respective conditional probabilities equal \( p_{ji} = q_{ji} \). Therefore, the SNE-algorithm should minimize the difference between the \( p_{ji} \) and \( q_{ji} \) probability distributions.

To measure the difference between probability distributions, one can use the Kullback-Leibler divergence, which in the analyzed case is identical to cross-entropy up to an additive constant. When working with the \( n \) points, it is necessary to minimize the sum of divergence:

\[ C = \sum_i KL(P_i \| Q_i) = \sum_i \sum_j p_{ji} \cdot \log_2 \frac{p_{ji}}{q_{ji}}, \]

where \( P_i \) and \( Q_i \) are conditional probability distribution of the \( x_i \) and \( y_i \) points.

The SNE-algorithm solves the problem of minimizing the sum of divergence with the help of classical gradient descent.

The gradient is represented as follows:

\[ \frac{\partial C}{\partial y_i} = 2 \cdot \sum_j (p_{ji} - q_{ji} + p_{ij} - q_{ij}) \cdot (y_i - y_j). \]

The Kullback-Leibler divergence is asymmetric: the enclosure of closely spaced points into distantly located points is associated with greater errors than the enclosure of distantly located points into closely located ones. The function (3) tends to preserve the local structure in a low-dimensional space (for acceptable dispersion measure \( \sigma_i \) in high-dimensional spaces).

The \( \sigma_i (i = 1, \ldots, n) \) measure of dispersion is the mean of the Gaussian density distribution of the \( x_i \) point. It is obvious that one and the same measure of dispersion cannot be optimal for all points of a high-dimensional space, for data are often unevenly distributed. It is logical to use lower dispersion \( \sigma_i \) for regions characterized by high density. Every value of dispersion \( \sigma_i \) for the \( x_i \) point triggers off the \( P_i \) probability distribution for other points in a high-dimensional space, whose entropy grows parallel to the growth of \( \sigma_i \).

The measure of the \( \sigma_i \) dispersion is calculated via binary search with perplexity \( Perp(P_i) = 2^{H(P_i)} \) as a priori parameter, where \( H(P_i) \) is the Shannon entropy of the \( P_i \) probability distribution measured in...
bites: \( H(P) = -\sum_j p_{ji} \cdot \log_2 p_{ji} \). Perplexity is interpreted as a measure of the effective number of neighbors.

When the SNE-algorithm is implemented, the \( y_i \) points are originally placed in a low-dimensional space in accordance with the Gaussian distribution with little dispersion and the mathematical expectation equalling zero. The function is then optimized (3).

It is believed that the function (3) is difficult to optimize. Moreover, the SNE-algorithm is characterized by high density. The t-SNE-algorithm attempts to solve both problems. Thus, the t-SNE-algorithm uses a symmetrical function of cost with simpler gradients, it also uses Student’s t-distribution instead of the Gaussian distribution to assess the similarity of two points in low-dimensional space. The t-SNE-algorithm uses a heavy-tailed distribution in a low-dimensional space to solve both the problem of high density and the problem of optimization.

Instead of minimizing the sum of the Kullback-Leibler divergence between the \( p_{ji} \) and \( q_{ji} \) conditional probabilities, one can minimize one Kullback-Leibler divergence between the simultaneous distribution of the \( P \) probabilities in a high-dimensional space and the simultaneous distribution of the \( Q \) probabilities in a low-dimensional space:

\[
C = KL(P \parallel Q) = \sum_i \sum_j p_{ji} \cdot \log_2 \frac{p_{ji}}{q_{ji}},
\]

where \( p_{ii} = 0; \ q_{ii} = 0 \).

Such an SNE-algorithm will be symmetrical, for \( p_{ji} = p_{ij}; \ q_{ji} = q_{ij} \) (regardless of the values of the \( I \) and \( j \) indexes). Therefore,

\[
q_{ij} = \frac{\exp(-\| y_i - y_j \|^2)}{\sum_k \exp(-\| y_k - y_i \|^2)}.
\]

Should we define \( p_{ij} \) as

\[
p_{ij} = \frac{\exp\left(-\frac{\| x_i - x_j \|^2}{2 \cdot \sigma^2}\right)}{\sum_k \exp\left(-\frac{\| x_k - x_i \|^2}{2 \cdot \sigma^2}\right)},
\]

it may cause a problem, is the \( x_i \) point is emitted (i.e. all \( \| x_i - x_j \|^2 \) are too great for the \( x_i \) point).

For such outliers of \( x_i \), the values of \( p_{ij} \) are really small for all the \( j \)-th points, therefore, the location of the \( y_i \) point produces a very little influence on the function (5). As a result, it is difficult to identify the location of the \( y_i \) relying on the location of other points of a low-dimensional space. The problem can be
solved if we define the $p_{ij}$ simultaneous probabilities in a high-dimensional space as symmetric conditional probabilities:

$$p_{ij} = \frac{p_{ji} + p_{ij}}{2 \cdot n}.$$  \hfill (7)

It guarantees that

$$\sum_j p_{ij} > \frac{1}{2 \cdot n} \quad \text{for any } x_i \text{ point. Then every } x_i \text{ point contributes to the function (5) significantly.}$$

In a low-dimensional space, the symmetric SNE-algorithm simply uses the formula (6).

The main distinction regarding the symmetric version of the SNE-algorithm is simpler gradients:

$$\frac{\partial C}{\partial y_i} = 4 \cdot \sum_j (p_{ij} - q_{ij}) \cdot (y_i - y_j),$$ \hfill (8)

which facilitates calculations.

3.2. The application of the t-SNE-algorithm for technical operation data visualization

The t-SNE-algorithm was used to visualize a data set containing information about the attributes of technical operations of parachute packing and landing platform mounting and mooring. The values of these attributes have been previously standardized.

The standardization procedure presupposes that every attribute is associated with the Gaussian distribution. When standardizing data, we change the scale of attribute value distribution in such a way so that the $m_j$ mean value for every $i$-th value ($i = 1, 245$) of the $j$-th attribute ($j = 1, 5$) equals 0, while the $\sigma_j$ standard deviation for every attribute equals 1. As a result, the values of every $j$-th attribute are centered and scaled:

$$y_{ij} = \frac{(x_{ij} - m_j)}{\sigma_j},$$

where $m_j = \frac{\sum_{i=1}^{245} x_{ij}}{245}$; $\sigma_j = \sqrt{\frac{\sum_{i=1}^{245} (x_{ij} - m_j)^2}{245}}$.

The analysis of the results of the original data visualization with different values of the t-SNE algorithm parameters shows the feasibility of using the DBSCAN-algorithm (Density-based spatial clustering of applications with noise) [10] to solve the task of technical operations clusterisation into a certain optimal (not predefined) number of clusters. Unlike other algorithms, the DBSCAN-algorithm does not require that the coordinates of cluster centers should be identified. More than that, clusters can be variously shaped.

4. Cluster analysis of technical operations data

4.1. Clusterisation using the DBSCAN-algorithm

The DBSCAN-algorithm solves the problem of data clusterisation with due consideration of point density in a metric space with the identification of the so-called outliers [10]. The DBSCAN-algorithm is used to
group all closely-located points (i.e. points with numerous close neighbours) together, labelling points that are located in low density regions (i.e. points that have no closely-spaced neighbours) as outliers.

The DBSCAN-algorithm presupposes that points to be clustered are divided into core points, which are located in high density regions, and outliers.

The $p$ point is a core point if there is a $\text{min}\_\text{sample}$ number of closely located points within the $\epsilon$ distance, which is the maximum radius of the $p$ point neighborhood. Such points can be reached directly from the $p$ point (data point).

The $q$ point is directly reachable from the $p$ point if the $q$ point is located within the $\epsilon$ distance of the $p$ point and the $p$ point is a core one.

The $A_{pq}$-point is reachable from the $p$ point if there is a chain of points $p_1, \ldots, p_n$, with $p_1 = p$ and $p_n = q$, where each $p_i$ point is reachable from the $p_i$ point (all points in the chain, except the $q$ point, should be core ones).

All points that are not reachable from core points are labelled as outliers.

If the $p$ point is a core point, it forms a cluster with all other points (both core points and outliers) that are reachable from it. Every cluster contains at least one core point. Outliers can be a part of a cluster, but they are positioned on the periphery, for they can’t be used to reach other points.

Reachability is not a symmetric ratio, for no point can be reached form an outlier whatever the distance may be.

The $p$ point and the $q$ point are density-connected if there is an $r$ point from which both the $p$ point and the $q$ point are reachable. The density-connection is symmetrical.

All points within a cluster are density-coupled. If a point is density-reachable from a certain point of a cluster, it belongs to the same cluster.

The main parameters of the DBSCAN-algorithm are the $\epsilon$ value, which identifies the maximum distance between two points that are considered neighbours, and the $\text{min}\_\text{sample}$ value, which identifies the number of points in a point’s neighbourhood (including the point itself) that are grouped together in a high density region making the point a core one.

The DBSCAN-algorithm selects an arbitrary point for which the $\epsilon$-neighbourhood is analyzed. If the $\epsilon$-neighbourhood contains fewer points than the $\text{min}\_\text{sample}$ number, they are used to create a cluster, otherwise the point is marked as an outlier. If an outlier enters the $\epsilon$-neighbourhood of another point, it is included into a respective cluster (provided the requirement of the number of points in the $\epsilon$-neighbourhood is fulfilled).

If a certain point is marked as a core point of a cluster, its $\epsilon$-neighbourhood will also be included into the same cluster.

The process of cluster formation continues until there appears a cluster that is not density-connected. After that a new point which has not been visited before is chosen and an attempt to form a cluster in its neighbourhood or classify it as an outlier is made.

The DBSCAN-algorithm can use any distance metric, though the Euclidian metric is the most common one.

The quality of the DBSCAN-algorithm clusterisation can be assessed with the help of the index of a cluster silhouette which should be maximised [19]. The optimal number of clusters is defined experimentally and is defined by variability results of the DBSCAN-algorithm parameters.

4.2. The application of the DBSCAN-algorithm for the clusterisation of technical operation data

When solving the task of cauterization of data describing technical operations, it is obvious that in order to maximize the clustering-related silhouette index, it is essential to accept the variability of attributes
assessed by experts through the prism of their knowledge and experience. Such attributes are the first, the second and the fifth.

However, the first attribute defines the type of operation and therefore needn’t vary (the type of operation is predefined and cannot be changed).

The second attribute, which identifies the complexity of technological equipment (tool) required for the performance of a technical operation, can vary: it can equal 4 if two or four tools are used simultaneously, it can be equal to 6 if complex equipment is required (clamp, molar strap, bodkin, etc.). If technical operations require simultaneous use of 2 or 4 tools, the value of the attribute will vary within the range of [3, 5]; if complex equipment is required, the attribute value varies within the range of [5, 7].

Speaking about the fifth attribute that defines the level of specialists’ professional competence required for the successful performance of a technical operation, one should remember that all values can vary. If a technical operation can be performed by a low-skilled specialist, the value will vary within the range of [1, 3]; if a technical operation can be performed by a middle-skilled worker, the attribute value will vary within the range of [4, 6]; if a technical operation requires a highly qualified specialist, the attribute values will vary within the range of [7, 9], see table 2.

Table 2. The first 12 lines of elementary actions with feature estimates and a calculated cluster label.

| Operations                        | Evaluating features | Cluster label |
|----------------------------------|---------------------|---------------|
| 1 Screwing in (screwing out) outriggers | 3 3 3 13 2 | 1             |
| 2 Attaching hooks to stern brackets | 3 3 1 4 2 | 1             |
| 3 Attaching corrugation devices to cutters | 4 5 2 22 2 | 2             |
| 4 Sealing leads                   | 3 3 1 1 2 | 1             |
| 5 Installing rear struts into grooves | 3 3 1 1 2 | 1             |
| 6 Installing rings into gears     | 3 3 1 1 2 | 1             |
| 7 Installing release actuators into brackets | 3 3 1 4 2 | 1             |
| 8 Screwing in (screwing out) chucks | 3 3 1 1 2 | 1             |
| 9 Adjusting with the help of a wrench | 8 7 0.5 2 2 | 3             |
| 10 Taking a line and holding it in the middle | 3 3 1 1 2 | 1             |
| 11 Opening hermetic seals with a special tool | 3 3 2 2 2 | 1             |
| 12 Inserting (extracting) a stud, a bolt, a screw, a pin | 3 3 2 28 2 | 1             |

The above described attribute value variability ensures a greater value of clustering-related silhouette index.

5. Experimental part
During the approbation of the proposed approach to the identification of the number and assortment of functional training solutions, the values of the second and the fifth attributes were arbitrary varied multiple times within a preset range, data sets containing information about the values of the five attributes of technical operations were, therefore, standardized. Each investigated data set was clustered using the DBSCAN-algorithm with various values of its attributes. The authors of the article used the software version of the DBSCAN-algorithm in the Python language (https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html). A grid search of such values of the DBSCAN-algorithm as the eps value and the min_sample value within the limits of the [0.15, 0.5] range (step size: 0.05) and the [3, 20] range (step size: 1), respectively, was performed. The optimal attribute values found by the grid search procedure are the following: eps=0.45, min_sample=3.

The authors of the article chose the variant which maximized the value of the cluster silhouette index equalling 0.7. All technical operations were grouped into 6 clusters (the number of functional training
solutions), which satisfied experts, who believed that the optimal number of training solutions should vary between 5 and 8.

The proposed training solutions were labelled as follows:
1. Parachute system canopy packing simulator,
2. Parachute system canopy packing and lines stowing simulator,
3. Binders and fasteners assembly and disassembly simulator,
4. Automatic activation device installation and deinstallation,
5. Cords and tapes knotting simulator,
6. Control and pre-flight inspection simulator.

The proposed simulators can be designed on the basis of extended-reality technologies (virtual, augmented, mixed). Extended reality is viewed as a novel universal component within the framework of methodological basis of industrial engineering, which will eventually become widely spread [20, 21].

Figure 1 shows the results of the visualization of a data set in a two-dimensional space using the t-SNE-algorithm created in the Python language (https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html).

Figure 1. Visualization of a technical-operation clustering procedure in a two-dimensional space.

All points are numbered to represent respective technical operations. Points marked by the DBSCAN-algorithm as belonging to different clusters are colored differently. Only one point in figure 1 is an outlier, which is not typical of the DBSCAN-algorithm. It indicates that the analyzed data is of high quality and the clusters are easily separable, while technical operations within them are characterized by compact character and high density. It should be noted that data (technical operations) clusterisation was performed within a high-dimensional space, while the visualization of points corresponding to the respective technical operations was performed within a low-dimensional space.
6. Conclusion

The article focuses on the authors’ approach to the solution of the task of the identification of the number and assortment of simulators, which is based on the t-SNE-algorithm and the DBSCAN-algorithm, which have never been used to solve such tasks. The results of the clustering procedure are intuitively clear in the majority of cases and enable one to ascertain whether a technical operation belongs to a certain cluster and, therefore, whether it can be used in some other simulators. The number of simulators proposed by the authors of the article equals 6, which is within the integral range [5, 8] suggested by experts.

References

[1] Yurt Sait N, Ozkol Ibrahim, Kaya Metin O and Haciyev Chingiz 2002 Optimization of the pd coefficient in a flight simulator control via genetic algorithms Aircraft Engineering and Aerospace Technology 2 pp 147-153
[2] George G and Kapadoukas Andrew Self 2000 A taxonomy of aircraft ground handling modes Aircraft Engineering and Aerospace Technology 4 pp 328-333
[3] Mayer G B Jr Determining the training effectiveness and cost effectiveness of visual flight simulators for military aircraft 1981 Master's Thesis Naval Post-Graduate School 23 pp 60-63 (Monterrey, CA)
[4] Brown D A Simulator aids aircraft 1972 Aviation Week and Space technology 6 pp 38-41
[5] Milligan G W and Cooper M C An Examination of Procedures for Determining the Number of Clusters in a Data Set 1985 Psychometrika Vol 50 pp 159-179
[6] Jain A.K and Dubes RC Algorithms for Clustering Data 1988 Prentice Hall p 334
[7] Peter H A and Sneath R S Numerical 1973 Taxonomy The principles and practice of numerical classification W.H. Freeman and Co p 573
[8] Bezdek JC and Ehrlich R Full W 1984 The fuzzy c-means clustering algorithm Computers & Geosciences Vol 10 (2-3) pp 191-203.
[9] Demidova L A and Nikulchev E E and Sokolova, Yu S 2015 Use of Fuzzy Clustering Algorithms’ Ensemble for SVM classifier Development International Review on Modelling and Simulations (IREMOS) 4 pp 446-457
[10] Ester M, Kriegel H-P, Sander J and Xu X 1996 A density-based algorithm for discovering clusters in large spatial databases with noise KDD’96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining pp 226-231
[11] Tenenbaum J B, de Silva V and Langford J C 2000 A global geometric framework for nonlinear dimensionality reduction Science Vol 290 (5500) pp 2319-2323.
[12] Roweis S T and Saul L K 2000 Nonlinear dimensionality reduction by Locally Linear Embedding Science Vol 290 (5500) pp 2323-2326
[13] Hinton G E and Roweis S T 2002 Stochastic Neighbor Embedding Advances in Neural Information Processing Systems Vol 15 pp 833-840 (USA)
[14] Belkin M and Niyogi P Laplacian 2002 Eigenmaps and spectral techniques for embedding and clustering Advances in Neural Information Processing Systems Vol 14 pp 585 – 591 (USA)
[15] Weinberger K Q and Sha F and Saul L K 2004 Learning a kernel matrix for nonlinear dimensionality reduction Proceedings of the 21st International Conference on Machine Learning
[16] Lee J A and Verleysen M 2007 Nonlinear dimensionality reduction Springer (USA)
[17] Laurens van der Maaten 2008 Visualizing Data using t-SNE Journal of Machine Learning Research, Vol 9 pp 2579-2605 (Toronto, ON, Canada)
[18] McInnes L and Healy J 2018 Uniform Manifold Approximation and Projection for Dimension Reduction *ArXiv e-prints* 1802.03426

[19] Rousseeuw P 1987 Silhouettes: a graphical aid to the interpretation and validation of cluster analysis Journal of Computational and Applied Mathematics Vol 20(1) pp 53-65 (Russian)

[20] Zuev A S, Makushchenko M A, Ivanov M E and Merkulov E S 2020 Extended reality technology – a new component in industrial engineering and production systems. *Russian Technological Journal* 8(4) 46-65 (In Russ.) https://doi.org/10.32362/2500-316X-2020-8-4-46-65

[21] Çöltekin Arzu, Lochhead Ian, Madden Marguerite, Christophe Sidonie, Devaux Alexandre, Pettit Christopher, Lock Oliver, Shukla Shashwat, Herman Lukáš, Stachoň Zdeněk, Kubiček Petr, Snopková Dajana, Bernardes Sergio and Hedley Nicholas 2020 Extended Reality in Spatial Sciences: A Review of Research Challenges and Future Directions *Int J Geo-Inf* 9(7) P 439