entropy

MDPI

Article

Small Stochastic Data Compactification Concept Justified in the Entropy Basis

Viacheslav Kovtun [1,*], Elena Zaitseva [2], Vitaly Levashenko [2], Krzysztof Grochla [1] and Oksana Kovtun [3]

[1] Internet of Things Group, Institute of Theoretical and Applied Informatics Polish Academy of Sciences, Bałtycka 5, 44-100 Gliwice, Poland; kgrochla@iitis.pl
[2] Department of Informatics, University of Žilina, 010 26 Žilina, Slovakia; elena.zaitseva@fri.uniza.sk (E.Z.); vitaly.levashenko@fri.uniza.sk (V.L.)
[3] Department of the Theory and Practice of Translation, Faculty of Foreign Languages, Vasyl' Stus Donetsk National University, 600-Richchya Str., 21, 21000 Vinnytsia, Ukraine; o.kovtun@donnu.edu.ua
* Correspondence: kovtun_v_v@vntu.edu.ua

Abstract: Measurement is a typical way of gathering information about an investigated object, generalized by a finite set of characteristic parameters. The result of each iteration of the measurement is an instance of the class of the investigated object in the form of a set of values of characteristic parameters. An ordered set of instances forms a collection whose dimensionality for a real object is a factor that cannot be ignored. Managing the dimensionality of data collections, as well as classification, regression, and clustering, are fundamental problems for machine learning. Compactification is the approximation of the original data collection by an equivalent collection (with a reduced dimension of characteristic parameters) with the control of accompanying information capacity losses. Related to compactification is the data completeness verifying procedure, which is characteristic of the data reliability assessment. If there are stochastic parameters among the initial data collection characteristic parameters, the compactification procedure becomes more complicated. To take this into account, this study proposes a model of a structured collection of stochastic data defined in terms of relative entropy. The compactification of such a data model is formalized by an iterative procedure aimed at maximizing the relative entropy of sequential implementation of direct and reverse projections of data collections, taking into account the estimates of the probability distribution densities of their attributes. The procedure for approximating the relative entropy function of compactification to reduce the computational complexity of the latter is proposed. To qualitatively assess compactification this study undertakes a formal analysis that uses data collection information capacity and the absolute and relative share of information losses due to compaction as its metrics. Taking into account the semantic connection of compactification and completeness, the proposed metric is also relevant for the task of assessing data reliability. Testing the proposed compactification procedure proved both its stability and efficiency in comparison with previously used analogues, such as the principal component analysis method and the random projection method.

Keywords: machine learning; data analysis; entropy; data reliability; small data; stochastic data; compactification; completeness; parametric optimization

Citation: Kovtun, V.; Zaitseva, E.; Levashenko, V.; Grochla, K.; Kovtun, O. Small Stochastic Data Compactification Concept Justified in the Entropy Basis. *Entropy* **2023**, *25*, 1567. https://doi.org/10.3390/e25121567

Academic Editor: Donald J. Jacobs

Received: 13 October 2023
Revised: 15 November 2023
Accepted: 18 November 2023
Published: 21 November 2023

1. Introduction

The most valuable resource in the information society is data. It seems that "there is no such thing as too much data", but let us try to look at this catchphrase as data scientists. The "curse of dimensionality" is a problem that consists of the exponential growth of the amount of data that has occurred simultaneously with the growth of the dimensionality of the space for data representation. This term was introduced by Richard Bellman in 1961. Scientists dealing with mathematical modelling and computational methods were the first to face this problem. Now, this problem is faced again as machine learning and artificial

intelligence methods are implemented. In this study, we will illustrate the relevance of this problem using the k-nearest neighbour method, which is popular for solving classification problems [1–4]. The essence of the method is as follows: the instance belongs to the same class as that which the majority of its nearest neighbour instances in the parametric space belong. To ensure high-quality work with this method, the saturation density of the parametric space with instances must be sufficiently high. How are the parametric space dimensions, the density of instances, and their number related to each other? To uniformly cover a unit interval [0, 1] with a density 0,01, we need 100 points, where the coverage density is defined as the ratio of the number of points evenly distributed in the target interval to the length of the latter. Now, imagine a 10-dimensional cube. To achieve the same coverage density, we already need 10^{20} points, that is, 10^{18} times more points compared to the original 1-dimensional space. This example demonstrates the reason for the inefficiency of the brute force method in typical machine learning problems (classification, clustering, and regression) [5–9]. The paradox is that it is impossible to solve the mentioned applied problems using a small number of parameters and achieve adequate results. One can simply turn a blind eye to the problem of dimensionality, which is the paradigm of deep learning, where using non-parameterized models achieves a significant increase in their quality despite the colossal increase in the number of calculations and accepting as an axiom the potential instability of the training process. But this recipe is unacceptable in the context of the machine learning ideology. The following Table 1 contains a more detailed comparison of these two methods.

Table 1. General comparison of the concepts of machine and deep learning.

Criterion	Machine Learning	Deep Learning
The number of data points	One can use small amounts of data to create forecasts	It is necessary to use large volumes of training data to create forecasts
Dependence on equipment	It can work on low-power computers. Large computing power is not required	Depends on high-performance computers. At the same time, the computer performs a large number of operations on the matrix. The graphic processor can effectively optimize these operations
The process of constructing features	Requires an accurate determination of the signs and their creation by users	Recognizes high levels based on data and independently creates new signs
Claim to training	The training process is divided into small steps. Then, the results of each step are combined into a single output block	The problem is solved by the method of thorough analysis
Training time	Training takes relatively little time, from a few seconds to several hours	As a rule, the training process takes a long time since the deep learning algorithm includes many levels
Output	The output data is usually a numerical value, for example, assessment or classification	The weekend can have several formats, such as text, estimate or sound

Therefore, managing the dimensionality of data while preserving their quality and the representativeness of the parametric space is an urgent scientific problem for machine learning.

The most widely used method for reducing data dimensionality is singular value decomposition (SVD, [10–12]). The matrices obtained as a result of SVD have a very specific interpretation in the machine learning methodology. They can be used according to the proven method both for principal component analysis (PCA, [13–15]) and (with certain reservations) for non-negative matrix factorization (NMF, [16–18]). SVD can also be used to improve the results of independent component analysis (ICA, [19–21]). It is convenient to apply SVD because there are no restrictions on the structure of the original data matrix (square when using the LU [22] or Schur distribution [23]; square, symmetric, or positive definite when using the Cholesky distribution [24]; matrix with positive elements when applying NMF). The essence of SVD is the representation of the original matrix X as a product of matrices of the form $X = U\Sigma V^*$, where U is a unitary matrix of order m and $\sum$ is

a rectangular diagonal matrix of dimension ($m \times n$), where m is a number of instances and n is a number of measured observables, with singular elements on the main diagonal and V^* is a matrix of order n, obtained as a result of conjugate transpose of the matrix V. The matrix $\sum$ is important for the dimensionality management problem. The squared singular elements of this matrix are interpreted as the variance σ^2 of the corresponding component. Based on the value of these variances, the researcher can select the required number of components. What is the best value $\sum_m \sigma^2$? Some recommend maintaining the inequality $\sum_m \sigma^2 \geq 0,90$, while others believe that $\sum_m \sigma^2 \geq 0,50$ is sufficient. The original answer to this question is provided by Horn's parallel analysis based on Monte Carlo simulation [25]. The disadvantage of both SVD and PCA is the high computational complexity of obtaining a singular distribution (well-known randomized algorithms [26] slightly mitigate this limitation). A more serious limitation is the sensitivity of SVD/PCA to outliers and the type of distribution of the original data. Most researchers believe that SVD/PCA works consistently with normally distributed data, but it has been empirically found that, as the data dimensionality increases, there are exceptions even to this rule. Therefore, SVD/PCA methods cannot guarantee the stability of the data dimensionality reduction procedure.

NMF is used to obtain the decomposition of a non-negative matrix $X_{(m \times n)}$ into non-negative matrices $W_{(m \times k)}$ and $H_{(k \times n)}$: $X = WH$. By choosing $k << m, n$, we can solve the problem of reducing the dimensionality of the original matrix quite effectively. The problem is that, unlike SVD, finding the $X = WH$ decomposition does not have an exact solution. There are specialized formulations of quadratic programming problems, such as the support vector machine (SVM, [27–29]) [30]. However, we understand that this means that NMF has the same limitations that have been pointed out for SVD/PCA.

The ICA method crossed into machine learning from the signal processing theory and, in its original formulation, was intended for the decomposition of a signal with additive components. At the same time, it was believed that these components have an abnormal distribution, and the sources of their origin are independent. To determine independent components, either minimization of mutual information based on Kullback–Leibler divergence [19] or minimization of "non-Gaussianity" [20,21] (using measures such as kurtosis coefficient and negentropy) are used. In the context of the dimensionality reduction problem, the application of ICA is trivial: to represent the input data as a mixture of components, divide them and select a certain number. There is no analytically consistent criterion for component selection.

We have often mentioned machine learning methods in the context of the data dimensionality management problem. However, there are competitors originating from the artificial intelligence field, i.e., the autoencoders [31–33]. This is an original class of neural networks, created so that the signal given to the input layer is reproduced as accurately as possible at the output of the neural network. The number of hidden layers should be at least one, and the activation functions of neurons on these layers should be non-linear (most often *sigmoid*, *tanh*, *ReLu*). If the number of neurons in the hidden layer is less than the number of neurons in the input layer, and we reproduce the input signal at the same time with sufficient accuracy as the output of the trained autoencoder, then the parameters of the neurons of the hidden layer are a compact approximate representation of the input signal. The advantage of this approach is that the neural network works for us. It is also very easy to orient the autoencoder to solve the data dimensionality increasing problem: it is sufficient that there are more neurons on the hidden layer than on the input layer. Disadvantages are also known: empirical search for the optimal configuration of the neural network (number of hidden layers, number of neurons on those layers, and selection of their activation functions), empirical selection of both the training algorithm and its parameters), and the neural network regularization methods (*L1*, *L2*, *dropout*). And we have not yet focused on the specific drawback of autoencoders, i.e., the tendency to degenerate hidden layers in the training process.

In recent years, there has been a growing interest in the research of data analysis, particularly within the context of regression analysis applied to inhomogeneous datasets. The existing research [34] explores the challenges presented by data that can be gathered from various sources or recorded at different time intervals, resulting in inherent inhomogeneities that complicate the process of regression modelling. The conventional framework of independent and identically distributed errors, typically associated with a single underlying model, is inadequate for handling such data. As the authors claim, traditional alternatives, like time-varying coefficients models or mixture models, can be computationally burdensome and impractical. So, the paper [34] proposes an aggregation technique based on normalized entropy (neagging) in contrast with such well-known aggregation procedures as bagging and magging. This approach has shown great promise, and the paper provides practical examples to illustrate its effectiveness using real-world datasets across various scenarios. However, the authors position their solution for working with large amounts of data or Big data. The issue of applicability of the mentioned procedures for compactification of small variable data has not been considered.

Taking into account the strengths and weaknesses of the mentioned methods, we will formulate the necessary attributes of scientific research.

The research object is the process of stochastic empirical data collection compactification.

The research subjects are probability theory and mathematical statistics, information theory, computational methods, mathematical programming methods, and experiment planning theory.

The research purpose is to formalize the process of finding the optimal probability distribution density of stochastic characteristic parameters of the empirical data compactification model with the maximum relative entropy between the original and compactified entities.

The research objectives are:

- formalize the concept of calculating the variable entropy estimation of the probability distribution density of the characteristic parameters of the stochastic empirical data collection;
- formalize the process of the stochastic empirical data collection compactification with the maximization of the relative entropy between the original and compactified entities;
- justify the adequacy of the proposed mathematical apparatus and demonstrate its functionality with an example.

The Motivation. One derives quantitative information on a class of objects by measuring a set of observables ("characteristic parameters") on a sample of objects taken from the class of interest. A set of values taken by the chosen observables on one of the objects is an instance. One of the basic problems in general data analysis is finding the optimal number of instances and the optimal (minimal) number of observables, that allow, in the presence of noise, to build regression models, estimate correlations between observables, and classify and cluster the objects in a machine learning approach. In this perspective, which is a very relevant one, the authors propose a model of noisy data based on a conditional, relative entropy [Equation (6)]. The article introduces a consistent and tunable method of "compactification" that performs quite well concerning other established methods, such as PCA and random projection methods.

2. Models and Methods

2.1. Statement of the Research

Let us characterize the researched process using a model in terms of linear programming, that is, by a function $z = f(v, w)$ that summarizes n weighted characteristic parameters $v \in \mathbb{R}^n$, where the weights w are interval stochastic values: $w \in W = \{w^- \leq w \leq w^+\}$, the properties of which are characterized by the probability distribution density $P(w)$.

Suppose that, as a result of m observations of the investigated process, empirical data with the structure $\langle V, y \rangle$ were obtained, where V is the training collection and each

empirical parametric vector $v^{(i)} = (v_{i1}, \ldots, v_{in}) \in V$, $v^{(i)} \in \mathbb{R}^n$, corresponds to an empirical initial value $y_i \in y$, $\forall i = \overline{1,m}$. When substituting data V into the model z, the equality of

$$z = \{z_i\} = Vw,\ i = \overline{1,m}, \tag{1}$$

must be fulfilled and which is provided by the training of the model z.

We consider that the values y_i of the original empirical vector y contain interference, which are represented by stochastic vector values $\varepsilon_i \in \varepsilon$, $i = \overline{1,m}$, $\varepsilon \in \mathrm{E} = \{\varepsilon^- \le \varepsilon \le \varepsilon^+\}$, with the probability density function $L(\varepsilon)$ of a stochastic vector ε. Taking into account interferences, we present expression (1) as

$$u = z + \varepsilon = V_{(m\times n)}w + \varepsilon, \tag{2}$$

where $u \in U = [u^-, u^+]$, $u^- = Vw^- + \varepsilon^-$, $u^+ = Vw^+ + \varepsilon^+$.

In the context of the formulated equation, the machine learning methodology is focused on determining the estimates $\widehat{P}(w)$ and $\widehat{L}(\varepsilon)$ of the corresponding probability distribution densities. The basis for this is model (2) and a set of empirical data V. Based on the known estimates of $\widehat{P}(w)$ and $\widehat{L}(\varepsilon)$, it is possible to outline the domain of stochastic vectors $u \in U$. Such a problem will be referred to as a *d*-problem. The authors devoted the article [35] directly to the solution of the *d*-problem.

On the other hand, the problem of compactification of the parametric space V of model (2) is solved by reducing the dimension of the characteristic parameters from n to r units, $r < n$, is also of practical value. Such a problem will be referred to as a *c*-problem.

Suppose that, as a result of the compactification of the original empirical data with the structure $\langle V, y\rangle$, a shortened parametric space $\mathbb{R}^r$ is obtained where each parametric vector $y^{(i)} = (v_{i1}, \ldots, v_{ir}) \in Y$, $y^{(i)} \in \mathbb{R}^r$, or $i = \overline{1,m}$, corresponds to the original interval stochastic value $a \in A = \{a^- \le a \le a^+\}$, $j = \overline{1,r}$, with the probability distribution density $A(a)$.

To describe compactified data $\langle Y, a\rangle$, we define the model

$$b = Y_{(m\times r)}a, a \in R^r, b \in R^m, \tag{3}$$

and the vector of observations is expressed as

$$s = b + \xi, \tag{4}$$

where the stochastic vector ξ is formed by interval values $\Xi = \{\xi^- \le \xi \le \xi^+\}$ with the probability distribution density $Z(\xi)$. The vectors s defined by expression (4) are interpreted as $S = [s^-, s^+]$, $s^- = Ya^- + \xi^-$, $s^+ = Ya^+ + \xi^+$.

Our further actions will be aimed at formulating:

- optimality criterion of the compactified data matrix $Y_{(m\times r)}$;
- a method for calculating the elements of the optimal compactified data matrix $Y_{(m\times r)}$;
- a method for comparing the probability distribution densities of outputs of models (2) and (4) as an indicator of the effectiveness of the proposed compactification concept.

2.2. The Concept of Entropy-Optimal Compactification of Stochastic Empirical Data

Let us focus on the analytical formalization of the entropic properties of empirical data, summarized by the matrix V. Let there be m independent instances in the collection of class X, each of which is characterized by the values of n attributes (characteristic parameters). The selection of instances in the collection X is random. In this context, the matrix X summarizes x_{ij}, $i = \overline{1,m}$, $j = \overline{1,n}$, stochastic attributes whose values are real numbers: $x_{ij} \ge 0$, $i = \overline{1,m}$, $j = \overline{1,n}$, satisfying the condition $\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} x_{ij} \le W$, where W is determined by the region of origin of instances of the class X.

We normalize the values of the elements of the matrix X relative to the selected scale with a resolution of Δ: $h_{ij} = \lceil x_{ij}/\Delta \rceil$, $i = \overline{1,m}$, $j = \overline{1,n}$, $\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} h_{ij} \le A \ge \lceil W/\Delta \rceil$. The step Δ is chosen to ensure sufficient variability of the resulting integer values of the stochastic elements of the matrix $H = (h_{ij})$, $i = \overline{1,m}$, $j = \overline{1,n}$.

Let us formalize the process of forming the values of the elements of the matrix H. Let us have A atomic units of the resource, which are distributed among $m \times n$ elements of the matrix H, and the probability of a resource unit falling into the element h_{ij} is characterized by the probability p_{ij}, $i = \overline{1,m}$, $j = \overline{1,n}$. The probability distribution of such a process is defined as

$$P(H) = A!\prod_{i=1}^{m}\prod_{j=1}^{n}\frac{p_{ij}^{h_{ij}}}{h_{ij}!}. \tag{5}$$

If the Moivre–Stirling approximation of factorials of large numbers is applied to the logarithmic representation of expression (5), we obtain an expression that characterizes the process described above based on the relative entropy:

$$E(H|P) = -\sum_{i=1}^{m}\sum_{j=1}^{n} h_{ij}\ln\frac{h_{ij}}{p_{ij}}, \tag{6}$$

where $P = (p_{ij})$, $i = \overline{1,m}$, $j = \overline{1,n}$.

Taking into account the proposed physical interpretation of the process of the matrix H values formation, it is appropriate to introduce such a characteristic parameter as the resource units a priori distribution, i.e., $V = (v_{ij} = p_{ij}A)$, $i = \overline{1,m}$, $j = \overline{1,n}$. Taking this parameter into account, expression (6) can be redefined as

$$E(H|V) \triangleq -\sum_{i=1}^{m}\sum_{j=1}^{n} h_{ij}\ln\frac{h_{ij}}{v_{ij}}. \tag{7}$$

Equality (7) is defined with accuracy up to the constant $A\ln A$. The essential connection between the sources of origin of the elements of the matrices X and H allows us to define the cross-entropy function as

$$E(X|V) \triangleq -\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}\ln\frac{x_{ij}}{v_{ij}}. \tag{8}$$

Based on expression (8), we write:

$$E(G|P) = -W\ln\frac{W}{A} - \sum_{i=1}^{m}\sum_{j=1}^{n} g_{ij}\frac{g_{ij}}{p_{ij}}, \tag{9}$$

where $g_{ij} = x_{ij}/W \in [0,1]$, $i = \overline{1,m}$, and $j = \overline{1,n}$, and the second term is the relative uncertainty characteristic of the stochastic matrix X.

Function (8) is concave for the entire range of values of the argument X and reaches a single extremum at the point $x_{ij}^{*} = v_{ij}/\mathrm{e}$, $\mathrm{e} = 2,718$, $i = \overline{1,m}$, $j = \overline{1,n}$. The extreme value of function (8) is equal to

$$E_{\max}(x^{*}|V) = \frac{1}{\mathrm{e}}\sum_{i=1}^{m}\sum_{j=1}^{n} v_{ij}. \tag{10}$$

The value (10) characterizes the maximum uncertainty of the matrix X for a defined matrix V. Let us emphasize other useful properties of function (8).

Let us define a matrix L with elements $l_{ij}(x_{ij}, v_{ij}) = \ln(x_{ij}/\mathrm{e}v_{ij})$, $i = \overline{1,m}$, and $j = \overline{1,n}$. Considering $V = L(X,V)$, expression (8) can be rewritten as

$$E(X|V) = E(X, L(X,V)) = -\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} l_{ij}\left(x_{ij}, v_{ij}\right) = \mathrm{Sp}\left(XL^{\mathrm{T}}(X,V)\right) = \mathrm{Sp}\left(L(X,V)X^{\mathrm{T}}\right), \quad (11)$$

where the symbols Sp and T represent the operations of trace finding and matrix transposition, respectively.

Based on the definition $l_{ij}(x_{ij}, v_{ij})$, we obtain the following inequality for the logarithmic function:

$$l_{ij}(x_{ij}, v_{ij}) \le (x_{ij} - v_{ij})/v_{\min},\ i = \overline{1,m},\ j = \overline{1,n}, \quad (12)$$

where $v_{\min} = \min\limits_{i,j} v_{ij}$.

Having transformed expression (11) and taking into account inequality (12), we determine the upper limit of cross entropy (8):

$$\hat{E}(X|V) = \mathrm{Sp}\left(XX^{\mathrm{T}}\right) - \mathrm{Sp}\left(XV^{\mathrm{T}}\right). \quad (13)$$

Function (13) is concave and follows all the properties of function (8).

Consider a non-degenerate $\left(\det\left(V^{T}_{(n\times m)}V_{(m\times n)}\right) \neq 0\right)$ matrix of empirical data $V_{(m,n)}$ with positive elements. Let us set the desired dimension of the parametric space: r, $r < n$, and enter into the matrix $Q = (q_{ij} \ge 0)$, $i = \overline{1,n}$, and $j = \overline{1,r}$. We obtain a direct projection of the matrix $Q_{(n\times r)}$ onto the parametric space R^{mr}: $Y_{(m\times r)} = V_{(m\times n)}Q_{(n\times r)}$. We obtain the inverse projection on the space R^{mn} using the matrix $S_{(r\times n)}$, and the values of all elements which are positive: $X_{(m\times n)} = V_{(m\times n)}Q_{(n\times r)}S_{(r\times n)}$. The dimensionality of both the obtained matrix X and the original matrix V is the same: $(m \times n)$.

Let us express the cross-entropy functional $E(X|V) = E\left(X_{(m\times n)}\middle|V_{(m\times n)}\right)$, taking into account the existence of the matrices $Q_{(n\times r)}$ and $S_{(r\times n)}$:

$$E(X|V) = E(Q,S|V) = E\left(Q_{(n\times r)}, S_{(r\times n)}\middle|V_{(m\times n)}\right) = -\sum_{i=1}^{m}\sum_{j=1}^{n} e_{ij}\left(Q_{(n\times r)}, S_{(r\times n)}\middle|V_{(m\times n)}\right), \quad (14)$$

where

$$e_{ij}\left(Q_{(n\times r)}, S_{(r\times n)}\middle|V_{(m\times n)}\right) = x_{ij}\left(Q_{(n\times r)}, S_{(r\times n)}\middle|V_{(m\times n)}\right)\ln\left(x_{ij}\left(Q_{(n\times r)}, S_{(r\times n)}\middle|V_{(m\times n)}\right)/v_{ij}\right),$$

$$x_{ij}\left(Q_{(n\times r)}, S_{(r\times n)}\middle|V_{(m\times n)}\right) = \sum_{k=1}^{r}\sum_{l=1}^{n} s_{kj}q_{lk}v_{il}, i = \overline{1,m}, j = \overline{1,n}.$$

The optimal configuration of the values of the positive matrices Q and S in the entropy basis is described by the expression

$$(Q^*, S^*) = \arg\max_{(Q,S)\ge 0} E(Q,S|V). \quad (15)$$

We will search for the extremum of the objective function (15) by the iterative gradient projection method [36,37], taking into account the need to cut off elements with negative values (observing condition $(Q,S) \ge 0$).

Let us analytically express the partial derivatives of the function $E(Q,S|V)$ in terms of the arguments, i.e., the elements of matrices Q and S:

$$\frac{\partial E(Q,S|V)}{\partial q_{kl}} = -\sum_{i=1}^{m}\sum_{j=1}^{n}\frac{\partial e_{ij}(Q,S|V)}{\partial x_{ij}}\frac{\partial x_{ij}(Q,S|V)}{\partial q_{kl}} \quad (16)$$

$$\frac{\partial E(Q,S|V)}{\partial s_{lh}} = -\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{\partial e_{ij}(Q,S|V)}{\partial x_{ij}}\frac{\partial x_{ij}(Q,S|V)}{\partial s_{lh}} \tag{17}$$

where $\partial e_{ij}(Q,S|X)/\partial x_{ij} = \ln\left(x_{ij}/v_{ij}\right)+1$, $\partial x_{ij}(Q,S|V)/\partial q_{kl} = s_{lj}v_{ik}$, $\partial x_{ij}(Q,S|V)/\partial s_{lh} = \sum\limits_{k=1}^{n} q_{kl}v_{ih}$, $i=\overline{1,m}$, $j=\overline{1,n}$, $k=\overline{1,n}$, $l=\overline{1,r}$, and $h=\overline{1,n}$.

Let us derive vectors $\vec{q}$ and $\vec{s}$ as a result vectorization of matrices Q and S, respectively. We identify the gradient vector of the relative entropy functional (14) with components (16) $\nabla_Q\left(\vec{q},\vec{s}\right)$. We identify the gradient vector $\nabla_S\left(\vec{q},\vec{s}\right)$ of the relative entropy functional (14) with components (17). We initialize the iterative procedure for finding the extremum of the objective function (15) based on the gradient projection method and in terms of the introduced entities.

For the 0th iteration, we take $X^{(0)}$, $V^{(0)}$, $\vec{q}^{(0)} > 0$, $\vec{s}^{(0)} > 0$.

For the nth iteration, we write:

$$\begin{gathered}
\vec{q}^{(n+1)} = \begin{cases} \vec{q}^{(n)} + \gamma_{\vec{q}}\nabla_Q\left(\vec{q}^{(n)},\vec{s}^{(n)}\right) \forall \vec{q}^{(n+1)} \geq 0, \\ \vec{q}^{(n)} \forall \vec{q}^{(n+1)} < 0, \end{cases} \\
\vec{s}^{(n+1)} = \begin{cases} \vec{s}^{(n)} + \gamma_{\vec{s}}\nabla_S\left(\vec{q}^{(n)},\vec{s}^{(n)}\right) \forall \vec{s}^{(n+1)} \geq 0, \\ \vec{s}^{(n)} \forall \vec{s}^{(n+1)} < 0, \end{cases} \\
\vec{q}^{(n+1)} \Rightarrow Q^{(n+1)}, \vec{s}^{(n+1)} \Rightarrow S^{(n+1)}, X^{(n+1)} = Q^{(n+1)}S^{(n+1)}V, \\
E^{(n+1)} = E\left(Q^{(n+1)},S^{(n+1)}|V\right) = \sum_{i=1}^{m}\sum_{j=1}^{r} x_{ij}^{(n+1)} \ln\frac{x_{ij}^{(n+1)}}{v_{ij}},
\end{gathered} \tag{18}$$

where parameters $\gamma_{\vec{q}}$, $\gamma_{\vec{s}}$ regulate increments in the corresponding dimension.

Iterative process (18) ends when the dynamics of the change in the value of the relative entropy functional becomes less than the threshold δ:

$$\delta_E = E^{(n+1)} - E^{(n)} = \frac{\mathrm{I}(V) - \mathrm{I}(Y(Q|V))}{\mathrm{I}(V)} \leq \delta, \tag{19}$$

where $\mathrm{I}(V) = \sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} v_{ij}\ln v_{ij}$ is the information capacity of the positive matrix $V_{(m\times n)}$. By analogy, we write: $\mathrm{I}(Y(Q|V)) = \sum\limits_{i=1}^{m}\sum\limits_{j=1}^{r} y_{ij}(Q|V)\ln y_{ij}(Q|V)$, where $y_{ij} = \sum\limits_{l=1}^{n} v_{il}q_{lj}$.

The computational complexity of the implementation of the iterative procedure just described increases nonlinearly with the increase in the dimension of the analyzed empirical matrices. Considering this circumstance, it is acceptable to define the elements of the matrix of the reduced dimension Q based on the approximately defined relative entropy functional $\widetilde{E}$. For example, let us use the approximation of the logarithmic function at the point $x_0 = w$: $\ln x < \ln w + (x-w)/w_{\min}$. For points $w = x_{ij}$ we find:

$$E(Q,S|V) \approx \widetilde{E}(Q,S|V) = \sum_{i=1}^{n}\sum_{j=1}^{m}\left(x_{ij}^2(Q,S|V) - x_{ij}(Q,S|V)v_{ij}\right). \tag{20}$$

Let us present the expression (20) in the matrix form:

$$\widetilde{E}(Q,S|V) = \mathrm{Sp}\left(XX^{\mathrm{T}}\right) - \mathrm{Sp}\left(XV^{\mathrm{T}}\right), = \dagger(X(Q,S),X(Q,S)) - \dagger(X(Q,S),V) \tag{21}$$

where the symbol $\dagger$ represents the Frobenius scalar product: $\mathrm{Sp}\left(AB^{\mathrm{T}}\right) = \mathrm{Sp}\left(BA^{\mathrm{T}}\right) = \dagger(A,B) = \dagger(B,A)$.

With a fixed matrix of empirical data V, we will minimize the functional $\widetilde{E}(Q,S|V)$ on the set of positive matrices Q and S:

$$\left(\widetilde{Q}^*,\widetilde{S}^*\right)=\arg\min_{(Q,S)\geq 0}\widetilde{E}(Q,S|V). \tag{22}$$

The procedure for finding $\left(\widetilde{Q}^*,\widetilde{S}^*\right)$ also uses components (16), and (17), which should be adapted to the scalar form of representation of the entities involved. Applying the rules of matrix differentiation to the functional (21), we obtain the following scalar interpretations of components (16), and (17):

$$\Delta_Q(Q,S)=\frac{\partial\widetilde{E}(Q,S|V)}{\partial X}\frac{\partial X}{\partial Q}=2S\mathrm{X}Q\mathrm{X}-S\mathrm{X}, \tag{23}$$

$$\Delta_S(Q,S)=\frac{\partial\widetilde{E}(Q,S|V)}{\partial X}\frac{\partial X}{\partial S}=2Q^{\mathrm{T}}\mathrm{X}Q\mathrm{X}-Q^{\mathrm{T}}\mathrm{X}, \tag{24}$$

where $\mathrm{X}=XX^{\mathrm{T}}$; $\Delta_Q(Q,S)$ and $\Delta_S(Q,S)$ are the gradients of matrices Q and S, respectively. The results of expressions (23), and (24) will be matrices of dimension $(n\times r)$.

We initialize the iterative procedure for finding the extremum of objective function (22) based on the gradient descent method and in terms of entities (23), and (24).

For the 0th iteration: we take $X^{(0)}$, $V^{(0)}$.

For the nth iteration, we write:

$$\begin{gathered}
Q^{(n+1)}=\begin{cases} Q^{(n)}+\gamma_Q\Delta_Q\widetilde{E}\left(Q^{(n)},S^{(n)}|V\right)\geq 0\forall Q^{(n+1)}\geq 0,\\ Q^{(n)}\forall Q^{(n+1)}<0,\end{cases}\\
S^{(n+1)}=\begin{cases} S^{(n)}+\gamma_S\Delta_S\widetilde{E}\left(Q^{(n)},S^{(n)}|V\right)\geq 0\forall S^{(n+1)}\geq 0,\\ S^{(n)}\forall S^{(n+1)}<0,\end{cases}\\
X^{(n+1)}=VQ^{(n+1)}S^{(n+1)},E^{(n+1)}=\sum_{i=1}^{m}\sum_{j=1}^{r}x_{ij}^{(n+1)}\ln\frac{x_{ij}^{(n+1)}}{v_{ij}}.
\end{gathered} \tag{25}$$

The iterative process (25) ends when the dynamics of the change in the value of the functional $\widetilde{E}(Q,S|V)$ becomes less than the set threshold δ: $E^{(n+1)}-E^{(n)}\leq\delta$.

In [35], the authors described the basic concept of solving the d- and c-problems mentioned in Section 2.1 for empirical data of the type V and Y, respectively. The result is the optimal probability distribution densities of characteristic parameters and interference (for the d-problem: $P^*(w)$, $L^*(\varepsilon)$, and for the c-problem: $A^*(a)$, $Z^*(\xi)$, respectively). The mathematical apparatus presented in Section 2.2 allows, based on linear models (2), and (4), to calculate normalized $U\cap S$ probability distributions $F_d\left(\vec{u}\right)$ and $F_c\left(\vec{s}\right)$ to determine the absolute difference between these functions in terms of relative entropy [38–41].

To preserve the integrity of the presentation of the material, we will demonstrate how the basic concept of solving the d-problem is implemented in the context of model (2). Let's define the functional $E(P(w),L(\varepsilon))$ on the probability distribution densities $P^*(w)$ and $L^*(\varepsilon)$. We need to solve the optimization problem with the following objective function and constraints:

$$\begin{gathered}
E(P(w),L(\varepsilon))=-\int_W P(w)\ln P(w)dw-\int_E L(\varepsilon)\ln L(\varepsilon)d\varepsilon\to\max\\
\int_W P(w)dw=1,\int_E L(\varepsilon)d\varepsilon=1,M\{z\}=\int_W VwP(w)dw+\int_E\varepsilon L(\varepsilon)d\varepsilon=y.
\end{gathered} \tag{26}$$

The solution to the optimization problem (26) in analytical form looks like

$$P^*(w)=\exp(-\theta,Vw)/\int_W\exp(-\theta,Vw)dw,L^*(\varepsilon)=\exp(-\theta,\varepsilon)/\int_B\exp(-\theta,\varepsilon)d\varepsilon,$$

where the Lagrange multipliers θ are determined as a result of solving the system of balance equations $M\{z\}$ in the interpretation $\int\limits_W VwP^*(w)dw + \int\limits_E \varepsilon L^*(\varepsilon)d\varepsilon = y$.

In the context of the model (2), the probability distribution density $F(u)$ of the observation vector u is defined as

$$F(u) = \int\limits_E \Pi(u-\varepsilon)L^*(\varepsilon)d\varepsilon = F_d(u), \tag{27}$$

where $F_d(u)$ is the desired probability distribution density of the d-problem model, and $\Pi(u-\varepsilon)$ is the density of the stochastic vector $u-\varepsilon$. From expression (27) we find $w = V^{\mathrm{T}}z/V^{\mathrm{T}}V$.

Considering the interval nature of the vector z: $z \in Z = [z^- = Vw^-, z^+ = Vw^+]$, we write $\eta(z) = P^*\left(V^{\mathrm{T}}z/V^{\mathrm{T}}V\right)$. Having normalized the function $\eta(z)$, we express the probability distribution density of the vector z as $\Pi(z) = \eta(z)/\int\limits_Z \eta(z)dz$.

To determine the probability distribution density $F_c(s)$ in the context of the model (4) (c-problem), it is necessary to repeat the sequence of actions embodied in expression (27) based on the empirical data matrix Y.

To compare the functions $F_d(u)$ and $F_c(s)$, it is necessary to normalize them on the common carrier $\Lambda = U \cap S$:

$$\widetilde{F}_d(\lambda) = F_d(\lambda)/\int\limits_\Lambda F_d(\lambda)d\lambda,\ \widetilde{F}_c(\lambda) = F_c(\lambda)/\int\limits_\Lambda F_c(\lambda)d\lambda. \tag{28}$$

To find the absolute share of information losses between functions $\widetilde{F}_d(\lambda)$ and $\widetilde{F}_c(\lambda)$ due to compaction Δ_E we define in terms of the relative entropy of RE as

$$RE\left(\widetilde{F}_d, \widetilde{F}_c\right) = \int\limits_\Lambda \widetilde{F}_c(\lambda)\ln\left(\widetilde{F}_c(\lambda)/\widetilde{F}_d(\lambda)\right),$$

$$\Delta_E = \frac{1}{2}RE\left(\widetilde{F}_d, \widetilde{F}_c\right) + RE\left(\widetilde{F}_d, \widetilde{F}_c\right). \tag{29}$$

Note that the minimum $\Delta_E = 0$ is reached at $\widetilde{F}_d(\lambda) = \widetilde{F}_c(\lambda)$.

3. Results

Let us begin the experimental Section by demonstrating the functionality of the mathematical apparatus proposed in Section 2.2 on a simple abstract example.

Suppose we have initial empirical data of the form $V_{(m=2\times n=2)} = \begin{pmatrix} 0,100 & 0,800 \\ 0,800 & 1,000 \end{pmatrix}$. In the context of model (2), we write $u = Vw + \varepsilon$. Suppose that $w \in W = [0,000; 5,000]$, $\varepsilon \in \mathrm{E} = [-0,500; 0,500]$. The output component is defined by the vector $y = (0,600; 1,400)$.

Let $r = 1$, then $Y_{(2\times 1)} = V_{(2\times 2)}Q_{(2\times 1)}$, where $Q_{(2\times 1)} = \begin{pmatrix} q_{11} \\ q_{21} \end{pmatrix}$ is the matrix for the direct projection. The compactification model (4) for the above values and conditions looks like this $s = Ya + \xi$, where $a \in a = [0,000; 5,000]$, $\xi \in \Xi = [-0,500; 0,500]$. The inverse projection operation is analytically characterized as $X_{(2\times 2)} = V_{(2\times 2)}Q_{(2\times 1)}S_{(1\times 2)}$, where $S_{(1\times 2)} = \begin{pmatrix} s_{11} & s_{12} \end{pmatrix}$ is the matrix for the inverse projection.

Our example is characterized by a small dimension, so we will use procedure (18) to determine the cross entropy. In this context, the cross entropy E between the original empirical matrix $V_{(2\times 2)}$ and the matrix $X_{(2\times 2)}$ obtained as a result of direct-inverse projection will be analytically determined by the expression $E = -\sum\limits_{i=1}^{2}\sum\limits_{j=1}^{2} x_{ij}\ln\frac{x_{ij}}{v_{ij}}$. The function

$E(Q,S)$ reaches an extremum at $Q^*_{\max} = \begin{pmatrix} 0,356 \\ 0,768 \end{pmatrix}$, $S^*_{\max} = \begin{pmatrix} 0,257 & 0,559 \end{pmatrix}$. Accordingly, the optimal compactified matrix Y has the form $Y^* = \begin{pmatrix} 0,356 \\ 0,768 \end{pmatrix}$.

The optimal probability distribution densities of the characteristic parameters w and interference ε for the matrix V defined at the beginning of the Section are characterized by the functions $P^*(w) = 1,221\exp(-0.888w_1 - 1,419w_2)$ and $L^*(\varepsilon) = 0,982\exp(-0,642\varepsilon_1 - 0,136\varepsilon_2)$. To compare the functions $F_d(u)$ and $F_c(s)$, it is necessary to normalize them on the common carrier, so, using (28), we find $0 \le \lambda_1 \le 1,778$, $0 \le \lambda_2 \le 3,842$. Then, with the defined functions (2), (4), and $P^*(w)$, the absolute share of information losses (29) of reducing the dimensionality of the space of characteristic parameters from n = 2 to r = 1 $\left(V_{(2\times 2)} \to Y^*_{(2\times 1)}\right)$ is equal to $\Delta_E = 0,245$, which allows us to consider the result of the proposed compactification procedure of the original empirical matrix V as adequate.

To prove the effectiveness of the proposed compactification method (18) (*Met*3), the method should be compared with popular analogues, namely, with the principal component analysis method (*Met*1) and the random projection method (*Met*2). Considering the linear nature of functions (2) and (4), we will experiment in the context of solving the verification problem (dichotomous classification) with a linear classifier. Let's formulate such a problem based on the terminology used.

We define the linear classifier model as

$$z(s_k) = \operatorname{sign}\left(\sum_{i=1}^{n} w_i v_i(s_k)\right) = \begin{cases} +1 \forall \sum\limits_{i=1}^{n} w_i v_i(s_k) \ge 0, \\ -1 \forall \sum\limits_{i=1}^{n} w_i v_i(s_k) < 0, \end{cases} \tag{30}$$

where $k \in \{1, m\}$ and the values of the weights $w \in R^n$ are unknown a priori.

Empirical data with the structure $\left\langle V_{(m\times n)}, y_{(m\times 1)} \right\rangle$ are available, and $y_k = \begin{cases} +1 \forall z(t_k) = +1, \\ -1 \forall z(t_k) = -1, \end{cases}$ where t_k is an instance of a class $\langle V, y \rangle$ with a number $k \in \{1, m\}$. The training of the classifier (30) is reduced to the minimization of the empirical risk function of the form $R(w) = \sum\limits_{i=1}^{m} \|y - z(w|V)\|^2$. To test the trained classifier (30), test empirical data with the structure $\left\langle U_{(l\times n)}, x_{(l\times 1)} \right\rangle$ were used.

The results of the classification $b(t_k) = sign\left(\sum\limits_{i=1}^{n} \hat{w}_i u_i(t_k)\right) = \{-1, 1\} \forall k = \overline{1, l}$ are synchronously compared with the corresponding elements of the vector x and taken into account in the form of the value of the function $I = \sum\limits_{k-1}^{l} \Delta(t_k)$, where $\Delta(t_k) = \begin{cases} 1 \forall b(t_k) = x(t_k), \\ 0 \forall b(t_k) \ne x(t_k). \end{cases}$. Accordingly, classification accuracy is defined as $\alpha = I/l$.

The conducted experiment consisted of solving the verification problem using classifier (30) for:

$e0$—basic empirical dataset $\left\langle V_{(m\times n)}, y_{(m\times 1)} \right\rangle + \left\langle U_{(l\times n)}, x_{(l\times 1)} \right\rangle$;

{$e1$, $e2$, $e3$}—the dataset $\langle V, y \rangle + \langle U, x \rangle$, the dimension of the attributes of the matrices V and U which underwent compactification from the initial n to the specified r elements by the method {*Met*1, *Met*2, *Met*3}.

The value r was iteratively reduced: $r = n - 1, n - 2, \ldots, 1$, forming a set of datasets at each of the stages {$e1$, $e2$} with the corresponding compactification degree. The number of compactification procedures $e3$ was determined by the set of threshold values (19).

For experiments, as necessary, tables of synthetic data of the required size were generated. For this, the *sklearn.datasets.make_classification*(*n_class* = 2, *n_clusters* = 2, *n_redundant* = 0, *class_sep* = 1.0, *n_informative* = {10, 15}) function of the *Python* programming language was

used. Before use, all generated data were normalized to fall within the unit interval [0, 1]. The experiments were carried out using *scipy.stats.bootstrap* cross-validation.

The algorithmic designs of the {*Met*1, *Met*2, *Met*3} methods were implemented by the functions of the *scipy* and *sklearn* libraries. The classifier (30) was implemented as a support vector machine with a linear kernel using the function *sklearn.svm.SVC*. The *Met*1 *Met*2 methods were implemented using the *sklearn.decomposition.PCA* and *sklearn.random_projection.GaussianRandomProjection* functions, respectively. The basis for the implementation of the author's method (18) was the *scipy.optimize.minimize* function (after inverting the objective function (15)). At the same time, the attribute *ftol* was considered to be related to the threshold (19).

As already mentioned in Section 2.2, the author's empirical data compactification method proposed in the form of procedure (18) is comparatively computationally complex (this is what prompted the authors to formalize the "simplified" iterative procedure (25)). However, *Met*1, *Met*2 analogues have their disadvantages, which appear when compacting large-dimensional data. For example, with a sufficiently large number of instances of data m and their heterogeneity, *Met*1 becomes unstable. We will conduct the first experiment of the form $\alpha = f(m, Met, r)$ for $m = \overline{5,10}$, $n = 10$, $r = \overline{10,5}$, Met = {*Met*1, *Met*2, *Met*3}. The obtained results are visualized in Figure 1.

The previous experiment characterized the ultra-compact empirical data compactification procedure: $m \approx n$, $n/2 \leq r \leq n$. Now, let us investigate how the verification accuracy α depends on the compactification of the initial data, for which $m >> r$, $m > n$. The experiments were carried out for two generated datasets $DS1$ and $DS2$. The first was characterized by dimension (m = 100, n = 10) and the second by dimension (10^4, 10^2). When processing the first dataset, we set r = {100, 90,. . ., 50} When working with the second dataset, we set r = {100, 90,. . ., 50} The obtained results are presented in Figure 2.

The following experiment is specific to *Met*3 because it concerns the detection of the dependence between the verification accuracy α and the dynamics of such parameters as the compactification degree r and the value of the threshold $\delta = \{0,5; 0,4; \ldots; 0,1\}$ (see expression (19)). To preserve the common information background, the remaining parameters were borrowed from the previous experiment without changes, namely: $DS = \{DS1, DS2\}$, $r(DS1)$ = {10, 9,. . ., 5}, and $r(DS2)$ = {100, 90,. . ., 50}. The resulting dependencies are visualized in Figure 3.

The empirical data compactification process is accompanied by an information loss. The absolute error as an indicator of information loss during compactification can be calculated by expression (29). The relative share of information losses during compactification can be calculated directly by expression (19) when implementing the compactification procedure (18). Figure 4 presents the calculated dependences of the relative share of information loss δ_E on the compactification method $Met = \{Met1, Met2, Met3\}$ for datasets $\{DS1, DS2\}$ with the corresponding ranges of changes in the compaction degree r.

Finally, we will conclude the Experimental Section with a study of *Met*3, the detection of the dependence between the relative share of information loss δ_E, and the dynamics of such parameters as the compactification degree r and the threshold value $\delta = \{0,5; 0,4; \ldots; 0,1\}$ (see expression (19)). To ensure a holistic perception of the material of the Section, the remaining parameters were borrowed from the previous experiment. The resulting dependencies are shown in Figure 5.

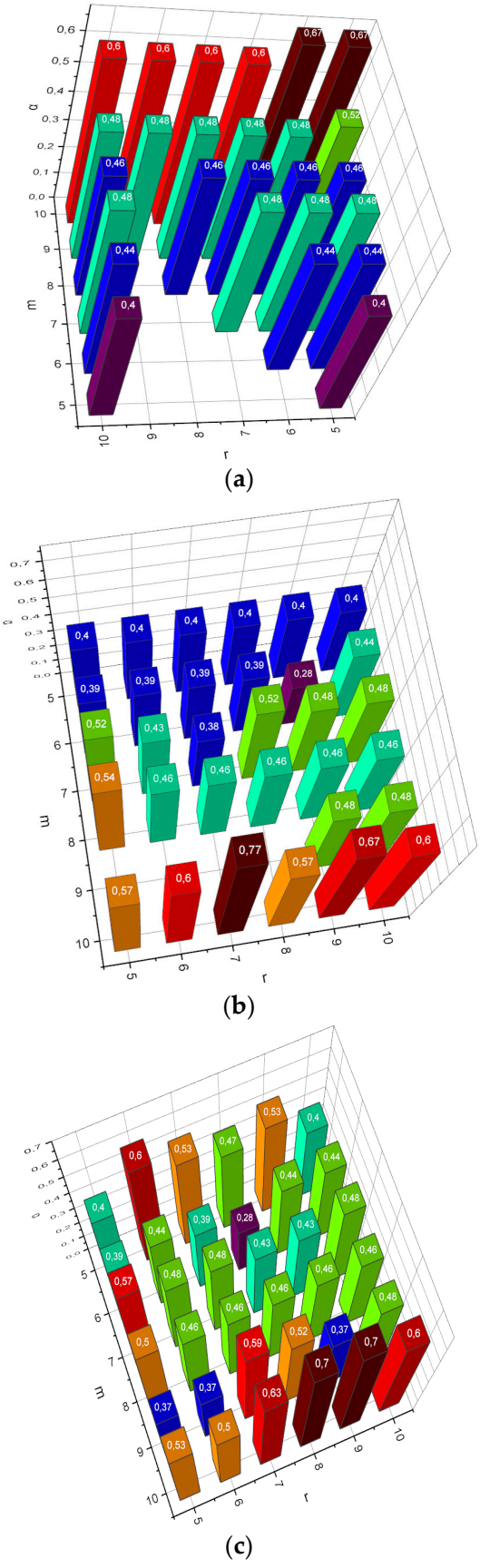


Figure 1. (**a**) Dependence $\alpha = f(m, Met1, r)$, $m = \overline{5,10}$, $r = \overline{10,5}$. (**b**) Dependence $\alpha = f(m, Met2, r)$, $m = \overline{5,10}$, $r = \overline{10,5}$. (**c**) Dependence $\alpha = f(m, Met3, r)$, $m = \overline{5,10}$, $r = \overline{10,5}$.

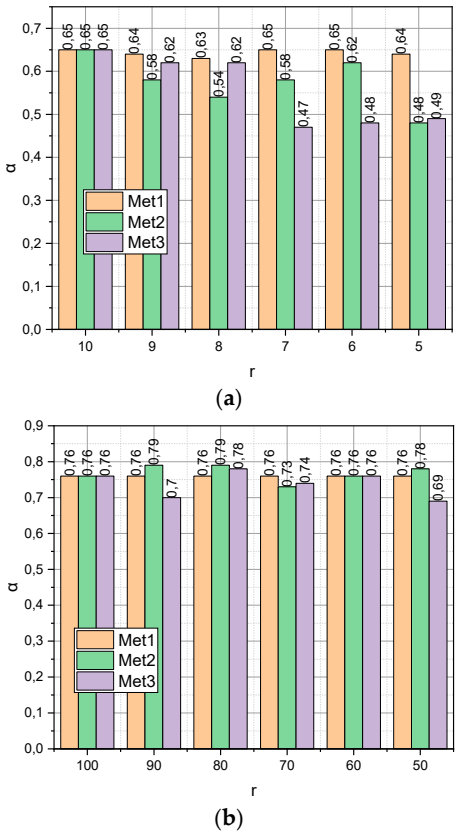


Figure 2. (**a**) Dependence $\alpha = f(Met, r, DS1)$, $r = \overline{10,5}$, $Met = \{Met1, Met2, Met3\}$. (**b**) Dependence $\alpha = f(Met, r, DS2)$, $r = \{100, 90, \ldots, 50\}$, $Met = \{Met1, Met2, Met3\}$.

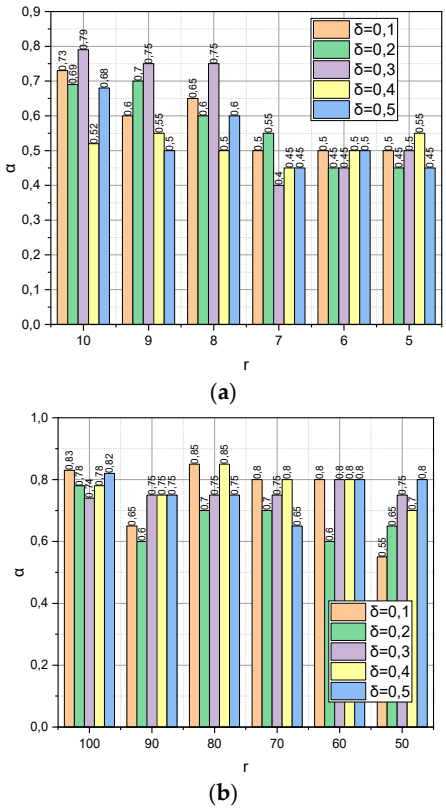
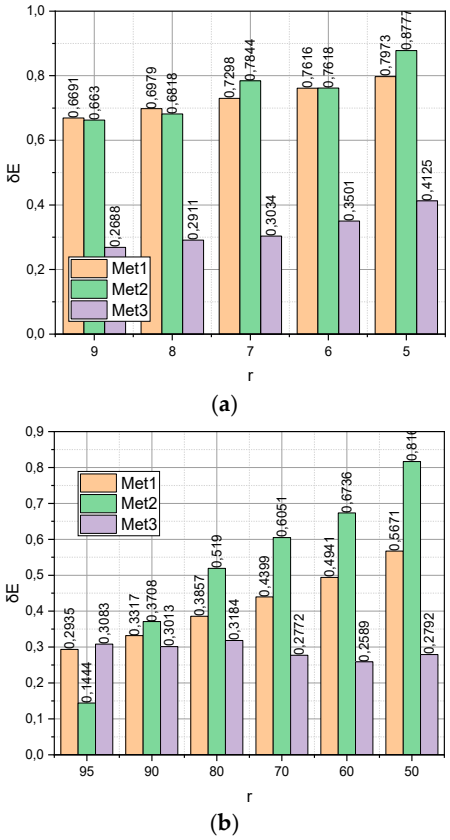


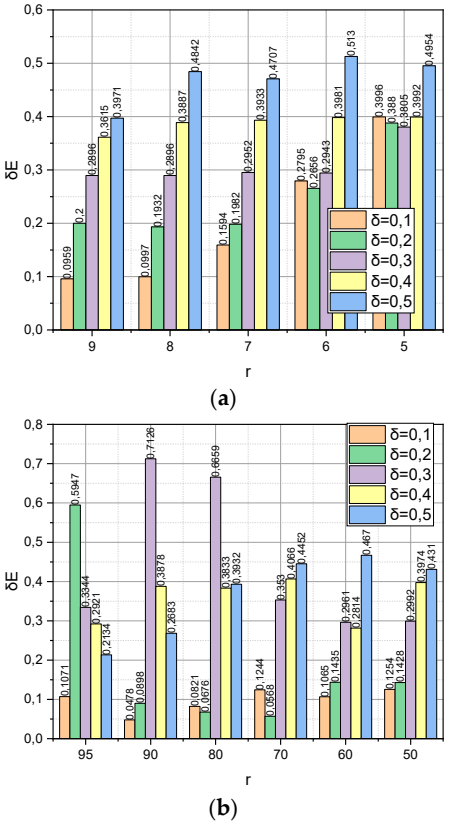
Figure 3. (**a**) Dependence $\alpha = f(r, \delta)$ for $DS1$ dataset. (**b**) Dependence $\alpha = f(r, \delta)$ for $DS2$ dataset.



(**a**)

(**b**)

Figure 4. (**a**) Dependence $\delta_E = f(r, Met)$ for *DS*1 dataset. (**b**) Dependence $\delta_E = f(r, Met)$ on *DS*2 dataset.



(**a**)

(**b**)

Figure 5. (**a**) Dependence $\delta_E = f(r, \delta)$ for *Met*3 and *DS*1 dataset. (**b**) Dependence $\delta_E = f(r, \delta)$ for *Met*3 and *DS*2 dataset.

4. Discussion

The research subject was chosen to reveal the characteristic features of the research object. This axiom works in all areas of science. Data analysis is no exception. There can be a huge, large, or small amount of data. The case with a small amount of data may be complicated by the fact that the source of the data, the process of its collection, or both, may not be under the researchers' control. In this case, data scientists will have to work with small stochastic data. The mathematical apparatus presented in Section 2.2 is focused on the problem of analyzing such data. Objective functions (15), and (22) implement the principle of maximum entropy formulated by Willard Gibbs in the context of compactification of (small) stochastic empirical data. Gibbs' work says that the most characteristic probability distributions of the states of an uncertain object are distributions that maximize the chosen measure of uncertainty, taking into account the available reliable information about the investigated object. The effectiveness of this approach is demonstrated by the results presented in Figure 1. Recall that, in this experiment, the compactification of extremely small data was carried out (the number of instances m in the data collection approached the number of attributes n). From Figure 1a,b, it can be seen that both the principal component analysis method ($Met1$) and the random projection method ($Met2$) demonstrated cases of non-functionality in situations when $m < r$, where r was the desired number of attributes in the compactified collection. The author's method ($Met3$) remained functional under any requirements determined by the experiment.

As shown in Figure 1, the results characterized the small empirical data compactification process: $m \approx n$, $n/2 \leq r \leq n$, then the results presented in Figure 2 show how the verification accuracy α depends on the compactification of the initial data, for which $m > n$ (a sufficient amount of empirical data, Figure 2a) or $m >> r$ ("big" empirical data, Figure 2b). From Figure 2a, it can be seen that $r \leq 7$ of function $\alpha(Met3)$ shows a monotonic linear character, in contrast to functions $\alpha(Met1)$ and $\alpha(Met2)$. This circumstance indicates that it was the author's method that made it possible to find the optimal configuration of the characteristic parameters space. Instead, the change of r in all functions $\alpha(Met)$ from Figure 2b is characterized by a non-linear character. It can also be seen that, with $r \leq 60$, it is the author's compactification method $Met3$ that generates the least informative parametric space in comparison with analogues. This fact can be explained by the fact that optimization method (18) does not have time to come close to the optimal distribution ensemble for the maximum number of iterations set of the algorithm (attribute $maxiter$= 1000 for the function $scipy.optimize.minimize$). The way out in such a situation can be the application of the approximate version of algorithm (18), represented by expressions (25).

Figure 3 demonstrates the dependence of the verification accuracy α on the dynamics of such parameters as the compactification degree r and the threshold value $\delta = \{0,5; 0,4; \ldots; 0,1\}$ (see expression (19)) of the completion of the iterative procedure (18). Let us notice that threshold δ is also a parameter that determines the maximum allowable reduction of the information capacity for the compactification data matrix. The usefulness of parameter δ lies in the fact that, based on its value, we can choose the permissible compaction degree r, not empirically (as, for example, in $Met1$) but analytically; if, after reducing the dimensionality of the dimension of the characteristic parameters to the value $r^{(n)}$, the estimate δ_E has decreased too much, then the compactification process should be stopped and the algorithm should be rolled back to the previous value of $r^{(n-1)}$. This is exactly the behaviour we observe in Figure 3a. Instead, as shown in Figure 3b, the situation is not stable. The probable explanation for this is similar to the one we mentioned regarding Figure 2b.

Figure 4 presents the calculated dependences of the relative share of information loss δ_E on the compactification method $Met = \{Met1, Met2, Met3\}$ for datasets $\{DS1, DS2\}$ with the corresponding ranges of changes in the compactification degree r. It can be seen that it is the function $\delta_E = f(r, Met3)$ with the growth r that grows significantly more slowly, surpassing competitors by almost two times. Note that this advantage was observed both for the "large" dataset $DS1$ and for the "Big" dataset $DS2$.

Figure 5 shows the relationship formalized by expression (19) between the relative share of information loss δ_E and the dynamics of such parameters as the compactification degree r and the threshold value δ. It is interesting that, for the dataset $DS1$ (Figure 5a), all values of r the condition $\delta_E \leq \delta$ are fulfilled, that is, algorithm (18) managed to find optimal distributions without exceeding the set limit on the permissible number of iterations. On the other hand, the circumstances were different for the "Big" dataset $DS2$. This can explain the unstable nature of the values presented in Figure 5b.

In general, the results presented in Section 3 prove both the functionality and the effectiveness of the mathematical apparatus presented in Section 2 in comparison with classical analogues, namely, the principal component analysis method and the random projection method. The obvious advantage of the author's method is the demonstrated stability of the small stochastic data compactification process and the possibility of analytical control of the loss of information capacity of the compactification data matrix. On the other hand, the disadvantage of the author's method is the computational complexity, which is especially evident when processing large data matrices. However, to mitigate this limitation, the authors propose an approximating simplified version (25) of the basic compactification procedure (18).

To implement the cross-entropy version of the author's compactification method, the method of conditional optimization on a non-negative orthant (CONNO) is adapted, and implemented in the *scipy* library. We note that, for some combinations of input data, the basic version of the CONNO method does not find a solution for the given optimization parameters. To test this concept, a series of experiments were adopted. The first series of experiments focused on identifying the dependence of classification accuracy on the number of objects (i.e., sample size). The study of this dependence for three compactification methods (PCA, RP, and author's) is important to identify areas of their application. It is known that entropy maximization methods and their derivatives, in particular the author's method, are usually used when the amount of data is limited compared to the dimension of the feature space. With "Big Data," there are no fundamental restrictions on their use, but computational difficulties increase significantly. The next series of experiments was focused on identifying the dependence of classification accuracy in conditions where the number of measurements significantly exceeds the number of characteristic parameters. The next series of experiments was focused on identifying the dependence of classification accuracy for the author's method on the acceptable reduction in the information capacity of the dataset. The next series of experiments was focused on assessing information losses from compactification implemented using and for the author's method. The experiments described above have already been carried out and results that positively characterize the author's method have been obtained. The problem is that, in its final form, the description, results obtained, and discussion are already more than 10 pages long. Increasing the size of this (already massive) article does not seem practical; therefore, if the mentioned experimental results interest you, dear reader, then I ask you to contact the corresponding author and he will be happy to share with you the results mentioned above.

5. Conclusions

Measurement is a typical way of gathering information about the investigated object, generalized by a finite set of characteristic parameters. The result of each iteration of the measurement is an instance of the class of the investigated object in the form of a set of values of characteristic parameters. An ordered set of instances forms a collection whose dimensionality for a real object is a factor that cannot be ignored. Managing the dimensionality of data collection, as well as classification, regression, and clustering, are fundamental problems of machine learning.

Compactification is the approximation of the original data collection by an equivalent collection (with a reduced dimension of characteristic parameters) with the control of accompanying information capacity losses. Related to compactification is the data completeness verifying procedure, which is characteristic of the data reliability assessment. If

there are stochastic parameters among the initial data collection characteristic parameters, the compactification procedure becomes more complicated. To take this into account, the research proposes a model of a structured collection of stochastic data defined in terms of relative entropy. The compactification of such a data model is formalized by an iterative procedure aimed at maximizing the relative entropy of sequential implementation of direct and reverse projections of data collections, taking into account the estimates of the probability distribution densities of their attributes. The procedure for approximating the relative entropy function of compactification to reduce the computational complexity of the latter is proposed. For a qualitative assessment of compactification, the metric of such indicators as the data collection information capacity, and the absolute and relative share of information losses due to compaction, are analytically formalized. Taking into account the semantic connection of compactification and completeness, the proposed metric is also relevant for the data reliability assessment task. Testing the proposed compactification procedure proved both its stability and efficiency in comparison with such used analogues as the principal component analysis method and the random projection method.

Further research is planned to attempt to simplify the procedure for finding entropy-optimal matrix projectors while observing the limit on permissible information losses from compactification.

Author Contributions: Conceptualization, V.K.; methodology, V.K.; software, V.K.; validation, E.Z., V.L., K.G. and O.K.; formal analysis, V.K.; investigation, V.K.; resources, E.Z., V.L., K.G. and O.K.; data curation, E.Z., V.L., K.G. and O.K.; writing—original draft preparation, V.K.; writing—review and editing, V.K., E.Z., V.L., K.G. and O.K.; visualization, V.K.; supervision, V.K.; project administration, V.K.; funding acquisition, V.K. All authors have read and agreed to the published version of the manuscript.

Funding: "Methodology for Increasing the Dependability of Information Systems for Critical Use with a Heterogeneous Wireless Interface", reg. no. 2022/45/P/ST7/03450, the POLONEZ BIS 2 program, implemented by the National Science Center in Krakow.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Most data are contained within the article. All the data are available on request due to restrictions, e.g., privacy or ethics.

Acknowledgments: The authors are grateful to all colleagues and institutions that contributed to the research and made it possible to publish its results.

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

References

1. Biswas, P.; Dandapat, S.K.; Sairam, A.S. Ripple: An approach to locate k nearest neighbours for location-based services. *Inf. Syst.* **2022**, *105*, 101933. [CrossRef]
2. Bansal, M.; Goyal, A.; Choudhary, A. A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning. *Decis. Anal. J.* **2022**, *3*, 100071. [CrossRef]
3. Izonin, I.; Tkachenko, R.; Dronyuk, I.; Tkachenko, P.; Gregus, M.; Rashkevych, M. Predictive modeling based on small data in clinical medicine: RBF-based additive input-doubling method. *Math. Biosci. Eng.* **2021**, *18*, 2599–2613. [CrossRef] [PubMed]
4. Izonin, I.; Tkachenko, R.; Shakhovska, N.; Lotoshynska, N. The Additive Input-Doubling Method Based on the SVR with Nonlinear Kernels: Small Data Approach. *Symmetry* **2021**, *13*, 612. [CrossRef]
5. Kamm, S.; Veekati, S.S.; Müller, T.; Jazdi, N.; Weyrich, M. A survey on machine learning based analysis of heterogeneous data in industrial automation. *Comput. Ind.* **2023**, *149*, 103930. [CrossRef]
6. Tymchenko, O.; Havrysh, B.; Tymchenko, O.O.; Khamula, O.; Kovalskyi, B.; Havrysh, K. Person Voice Recognition Methods. In Proceedings of the 2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP), Lviv, Ukraine, 21–25 August 2020; IEEE: Piscataway, NJ, USA, 2020. [CrossRef]
7. Bisikalo, O.; Kovtun, O.; Kovtun, V.; Vysotska, V. Research of Pareto-Optimal Schemes of Control of Availability of the Information System for Critical Use. In Proceedings of the 2020 1st International Workshop on Intelligent Information Technologies & Systems of Information Security (IntelITSIS), Khmelnytskyi, Ukraine, 10–12 June 2020; CEUR-WS. Volume 2623, pp. 174–193.

8. Bisikalo, O.V.; Kovtun, V.V.; Kovtun, O.V.; Danylchuk, O.M. Mathematical Modeling of the Availability of the Information System for Critical Use to Optimize Control of its Communication Capabilities. *Int. J. Sens. Wirel. Commun. Control.* **2021**, *11*, 505–517. [CrossRef]
9. Bisikalo, O.; Danylchuk, O.; Kovtun, V.; Kovtun, O.; Nikitenko, O.; Vysotska, V. Modeling of Operation of Information System for Critical Use in the Conditions of Influence of a Complex Certain Negative Factor. *Int. J. Control. Autom. Syst.* **2022**, *20*, 1904–1913. [CrossRef]
10. Bisikalo, O.; Bogach, I.; Sholota, V. The Method of Modelling the Mechanism of Random Access Memory of System for Natural Language Processing. In Proceedings of the 2020 IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), Lviv-Slavske, Ukraine, NJ, USA, 25–29 February 2020; IEEE: Piscataway, NJ, USA. [CrossRef]
11. Mochurad, L.; Horun, P. Improvement Technologies for Data Imputation in Bioinformatics. *Technologies* **2023**, *11*, 154. [CrossRef]
12. Stankevich, S.; Kozlova, A.; Zaitseva, E.; Levashenko, V. Multivariate Risk Assessment of Land Degradation by Remotely Sensed Data. In Proceedings of the 2023 International Conference on Information and Digital Technologies (IDT), Zilina, Slovakia, 20–22 June 2023. [CrossRef]
13. Kharchenko, V.; Illiashenko, O.; Fesenko, H.; Babeshko, I. AI Cybersecurity Assurance for Autonomous Transport Systems: Scenario, Model, and IMECA-Based Analysis. In *Communications in Computer and Information Science*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 66–79. [CrossRef]
14. Izonin, I.; Tkachenko, R.; Krak, I.; Berezsky, O.; Shevchuk, I.; Shandilya, S.K. A cascade ensemble-learning model for the deployment at the edge: Case on missing IoT data recovery in environmental monitoring systems. *Front. Environ. Sci.* **2023**, *11*, 1295526. [CrossRef]
15. Auzinger, W.; Obelovska, K.; Dronyuk, I.; Pelekh, K.; Stolyarchuk, R. A Continuous Model for States in CSMA/CA-Based Wireless Local Networks Derived from State Transition Diagrams. In *Proceedings of International Conference on Data Science and Applications*; Springer: Singapore, 2021; pp. 571–579. [CrossRef]
16. Deng, P.; Li, T.; Wang, D.; Wang, H.; Peng, H.; Horng, S.-J. Multi-view clustering guided by unconstrained non-negative matrix factorization. *Knowl.-Based Syst.* **2023**, *266*, 110425. [CrossRef]
17. De Handschutter, P.; Gillis, N.; Siebert, X. A survey on deep matrix factorizations. *Comput. Sci. Rev.* **2021**, *42*, 100423. [CrossRef]
18. De Clercq, M.; Stock, M.; De Baets, B.; Waegeman, W. Data-driven recipe completion using machine learning methods. *Trends Food Sci. Technol.* **2016**, *49*, 1–13. [CrossRef]
19. Shu, L.; Lu, F.; Chen, Y. Robust forecasting with scaled independent component analysis. *Finance Res. Lett.* **2023**, *51*, 103399. [CrossRef]
20. Moneta, A.; Pallante, G. Identification of Structural VAR Models via Independent Component Analysis: A Performance Evaluation Study. *J. Econ. Dyn. Control.* **2022**, *144*, 104530. [CrossRef]
21. Zhang, R.; Dai, H. Independent component analysis-based arbitrary polynomial chaos method for stochastic analysis of structures under limited observations. *Mech. Syst. Signal Process.* **2022**, *173*, 109026. [CrossRef]
22. HLi, H.; Yin, S. Single-pass randomized algorithms for LU decomposition. *Linear Algebra its Appl.* **2020**, *595*, 101–122. [CrossRef]
23. Iwao, S. Free fermions and Schur expansions of multi-Schur functions. *J. Comb. Theory Ser. A* **2023**, *198*, 105767. [CrossRef]
24. Terao, T.; Ozaki, K.; Ogita, T. LU-Cholesky QR algorithms for thin QR decomposition. *Parallel Comput.* **2020**, *92*, 102571. [CrossRef]
25. Trendafilov, N.; Hirose, K. Exploratory factor analysis. In *International Encyclopedia of Education*, 4th ed.; Elsevier: Amsterdam, The Netherlands, 2023; pp. 600–606. [CrossRef]
26. Fu, Z.; Xi, Q.; Gu, Y.; Li, J.; Qu, W.; Sun, L.; Wei, X.; Wang, F.; Lin, J.; Li, W.; et al. Singular boundary method: A review and computer implementation aspects. *Eng. Anal. Bound. Elements* **2023**, *147*, 231–266. [CrossRef]
27. Roy, A.; Chakraborty, S. Support vector machine in structural reliability analysis: A review. *Reliab. Eng. Syst. Saf.* **2023**, *233*, 109126. [CrossRef]
28. Çomak, E.; Arslan, A. A new training method for support vector machines: Clustering k-NN support vector machines. *Expert Syst. Appl.* **2008**, *35*, 564–568. [CrossRef]
29. Chen, H.L.; Yang, B.; Wang, S.J.; Wang, G.; Liu, D.Y.; Li, H.Z.; Liu, W.B. Towards an optimal support vector machine classifier using a parallel particle swarm optimization strategy. *Appl. Math. Comput.* **2014**, *239*, 180–197. [CrossRef]
30. Pineda, S.; Morales, J.M.; Wogrin, S. Mathematical programming for power systems. In *Encyclopedia of Electrical and Electronic Power Engineering*; Elsevier: Amsterdam, The Netherlands, 2023; pp. 722–733. [CrossRef]
31. Li, P.; Pei, Y.; Li, J. A comprehensive survey on design and application of autoencoder in deep learning. *Appl. Soft Comput.* **2023**, *138*, 110176. [CrossRef]
32. Mishra, D.; Singh, S.K.; Singh, R.K. Deep Architectures for Image Compression: A Critical Review. *Signal Process.* **2022**, *191*, 108346. [CrossRef]
33. Zheng, J.; Qu, H.; Li, Z.; Li, L.; Tang, X. A deep hypersphere approach to high-dimensional anomaly detection. *Appl. Soft Comput.* **2022**, *125*, 109146. [CrossRef]
34. Costa, M.C.; Macedo, P.; Cruz, J.P. Neagging: An Aggregation Procedure Based on Normalized Entropy. In Proceedings of the International Conference Of Numerical Analysis And Applied Mathematics ICNAAM 2020, Crete, Greece, 19–25 September 2022. [CrossRef]

35. Bisikalo, O.; Kharchenko, V.; Kovtun, V.; Krak, I.; Pavlov, S. Parameterization of the Stochastic Model for Evaluating Variable Small Data in the Shannon Entropy Basis. *Entropy* **2023**, *25*, 184. [CrossRef]
36. Zeng, Z.; Ma, F. An efficient gradient projection method for structural topology optimization. *Adv. Eng. Softw.* **2020**, *149*, 102863. [CrossRef]
37. El Masri, M.; Morio, J.; Simatos, F. Improvement of the cross-entropy method in high dimension for failure probability estimation through a one-dimensional projection without gradient estimation. *Reliab. Eng. Syst. Saf.* **2021**, *216*, 107991. [CrossRef]
38. Liu, B.; Chai, Y.; Huang, C.; Fang, X.; Tang, Q.; Wang, Y. Industrial process monitoring based on optimal active relative entropy components. *Measurement* **2022**, *197*, 111160. [CrossRef]
39. Fujii, M.; Seo, Y. Matrix trace inequalities related to the Tsallis relative entropies of real order. *J. Math. Anal. Appl.* **2021**, *498*, 124877. [CrossRef]
40. Makarichev, V.; Kharchenko, V. Application of Dynamic Programming Approach to Computation of Atomic Functions. In *Radioelectronic and Computer Systems*; no. 4; National Aerospace University-Kharkiv Aviation Institute: Kharkiv, Ukraine, 2021; pp. 36–45. [CrossRef]
41. Dotsenko, S.; Illiashenko, O.; Kharchenko, V.; Morozova, O. Integrated Information Model of an Enterprise and Cybersecurity Management System. *Int. J. Cyber Warf. Terror.* **2022**, *12*, 1–21. [CrossRef]