mathematics

MDPI

Article

Explainable Deep Learning: A Visual Analytics Approach with Transition Matrices

Pavlo Radiuk [1,*], Olexander Barmak [1], Eduard Manziuk [1] and Iurii Krak [2,3]

[1] Department of Computer Science, Khmelnytskyi National University, 11 Instytuts'ka Str., 29016 Khmelnytskyi, Ukraine; barmako@khmnu.edu.ua (O.B.); manziuk.e@khmnu.edu.ua (E.M.)
[2] Department of Theoretical Cybernetics, Taras Shevchenko National University of Kyiv, 4d Akademika Glushkova Ave, 03680 Kyiv, Ukraine; iurii.krak@knu.ua
[3] Laboratory of Communicative Information Technologies, V.M. Glushkov Institute of Cybernetics, 40 Akademika Glushkova Ave, 03187 Kyiv, Ukraine
* Correspondence: radiukp@khmnu.edu.ua; Tel.: +38-(097)-854-9148

Abstract: The non-transparency of artificial intelligence (AI) systems, particularly in deep learning (DL), poses significant challenges to their comprehensibility and trustworthiness. This study aims to enhance the explainability of DL models through visual analytics (VA) and human-in-the-loop (HITL) principles, making these systems more transparent and understandable to end users. In this work, we propose a novel approach that utilizes a transition matrix to interpret results from DL models through more comprehensible machine learning (ML) models. The methodology involves constructing a transition matrix between the feature spaces of DL and ML models as formal and mental models, respectively, improving the explainability for classification tasks. We validated our approach with computational experiments on the MNIST, FNC-1, and Iris datasets using a qualitative and quantitative comparison criterion, that is, how different the results obtained by our approach are from the ground truth of the training and testing samples. The proposed approach significantly enhanced model clarity and understanding in the MNIST dataset, with SSIM and PSNR values of 0.697 and 17.94, respectively, showcasing high-fidelity reconstructions. Moreover, achieving an F_1m score of 77.76% and a weighted accuracy of 89.38%, our approach proved its effectiveness in stance detection with the FNC-1 dataset, complemented by its ability to explain key textual nuances. For the Iris dataset, the separating hyperplane constructed based on the proposed approach allowed for enhancing classification accuracy. Overall, using VA, HITL principles, and a transition matrix, our approach significantly improves the explainability of DL models without compromising their performance, marking a step forward in developing more transparent and trustworthy AI systems.

Keywords: explainable artificial intelligence (XAI); deep learning; machine learning; visual analytics; human-in-the-loop; model explainability; transition matrix

MSC: 68T20

Citation: Radiuk, P.; Barmak, O.; Manziuk, E.; Krak, I. Explainable Deep Learning: A Visual Analytics Approach with Transition Matrices. *Mathematics* **2024**, *12*, 1024. https://doi.org/10.3390/math12071024

Academic Editors: Mohamed A. Sharaf and Kostas Stefanidis

Received: 15 February 2024
Revised: 25 March 2024
Accepted: 27 March 2024
Published: 29 March 2024

1. Introduction

In this research paper, we explore the critical challenges posed by the non-transparent nature of artificial intelligence (AI) systems, particularly in the domain of deep learning (DL), and the imperative for explainable artificial intelligence (XAI) to make these systems comprehensible to humans [1]. The concept of "black box" AI highlights systems whose internal decision-making processes are not transparent, leaving developers and end users alike unable to understand or trust their operations. This lack of transparency directly contrasts with "white box" models [2], which are designed to be interpretable by experts, thus ensuring a level of trustworthiness and comprehension in their outcomes.

XAI emerges as a solution to the "black box" problem by facilitating systems that can articulate their decision-making process in human-understandable terms [3]. This is not

just a technical enhancement but a fulfillment of a societal right—the right to explanations that are clear and justifiable, especially when decisions have significant impacts [4,5]. The legal and ethical discourse around this underscores its importance in an increasingly digital society that relies on AI and machine learning (ML) technologies [6–8].

Our research employs the concept of mental models (MMs) and formal models (FMs) to bridge the gap between human understanding and AI decision-making processes. MMs represent the intuitive understanding humans have of the world, influencing our cognition and decision-making [9]. FMs, however, are derived from AI methods and are often not intuitively understandable to humans [10]. By transitioning from FM to MM, XAI aims to correct misconceptions and enhance user performance, aligning with the social right to an explanation [11].

In this study, we also delve into the principles of transparency, interpretation, and explanation in XAI methods, highlighting the need for transparent processes [3], interpretable models [12], and clear explanations of decisions [13]. The challenges in achieving this with DL methods, which often surpass traditional ML in performance but lack explainability, are acknowledged [14,15]. Explanation is considered a key aspect, though its term is not exhaustively defined [13]. It is believed that XAI can be considered as "a set of characteristics of the interpreted domain that contributed to the decision-making in a specific case" [16] (for example, regression [17] or classification [18]). Methods that meet these criteria are considered capable of providing a rationale for the decisions made, control, and thus their verification, improvement, and learning of new facts [19]. This brings to light concerns around biases and the ethical implications encapsulated in the FACTS framework—Fairness, Accountability, Confidentiality, Transparency, and Safety [20].

To address the above-mentioned challenges, visual analytics (VA) and the human-in-the-loop (HITL) principle are employed. VA leverages visual representations to simplify complex data analyses, revealing patterns and trends [21,22]. Visualizing data in the form of graphs, charts, and maps helps to identify patterns and contributes to the development of effective ML and DL models. Visualization aids in uncovering patterns, trends, and correlations that might otherwise go unnoticed. Notably, multidimensional scaling (MDS) [23] is considered an effective VA tool, allowing for the visualization of multidimensional data based on their similarity, presenting information about the distances between objects as points in an abstract space. Furthermore, the concept of HITL is used to denote human participation in the process of aiding the computer to make accurate decisions during model development [24]. HITL enhances the ML process compared to random sampling by selecting the most important data to achieve the best results [25,26].

In the dynamic landscape of DL, the rise of complex models requires utilizing methods that not only make accurate predictions but also explain how these models arrive at those predictions. This need for explainability comes from having to understand the reasoning behind a model's decisions, which is critical in areas where explaining the model's judgments is necessary to validate its usefulness and trustworthiness, e.g., healthcare, media, legal field, etc. In this regard, our research employs VA and HITL approaches to improve the explainability of DL models. We focus on exploring different representations of the complex DL models, each capturing distinct features or characteristics, to explain how they (as FMs) make predictions.

Our study is based on the hypothesis that a single dataset could lead to multiple interpretable models (or feature sets), each depicted by distinct feature matrices. Within our research, we posit that the number of rows in these matrices should correspond to the number of samples in the training set, which lies between the number of classes in the sample and the total number of training samples. We also suggest that these different representations can be connected through a method that describes the transformations between them. For our purposes, we analyze both comprehensive holistic models as well as dimensionally reduced versions. Specifically, we delve into scenarios like comparing a DL model with its ML counterpart or connecting a multidimensional data model with a reduced-dimensional feature space model. A key challenge this study addresses is the

difficulty in grasping the physical significance of features in formal DL models, which often remain elusive to human understanding.

Overall, the goal of this study is to improve DL explainability through VA and HITL. To achieve this goal, we propose a novel approach that involves constructing a transition matrix to enable seamless mapping between two distinct feature spaces: the formal models employed in DL and the mental models presented as an ML model that resonates more intuitively with human cognition. The key contributions of this research include:

- A novel method that employs a transition matrix to transform the outcomes of a formal DL model into a format that is more intuitive for human understanding, i.e., a mental model, by leveraging an ML model with features that are more easily interpreted by humans.
- An improved method for obtaining a separating hyperplane for classification tasks using a transition matrix based on VA and the HITL principle.

The ability of the proposed methods to solve the specified tasks is demonstrated through numerical examples and validated with benchmark datasets. It should be also clarified that the possibility of empirical validation, i.e., evaluation of the effectiveness of the proposed approach compared to analogs, depends on specific subject areas and tasks. We believe that the quality of any interpretation is a subjective indicator and depends on the end user's ability to understand the decisions made by DL models. If the proposed interpretation method allows an end user to trust the decision made by the DL model, this, in our opinion, is sufficient for assessing the quality of the proposed approach.

The structure of the article is as follows. Section 2 "Basic Definitions and Concepts" presents the main theoretical definitions and concepts used in the article. Section 3 "Materials and Methods" contains a theoretical description of the proposed methods for explaining results obtained by the DL model through the ML model and obtaining coefficients of the hyperplane for points belonging to this hyperplane. The theoretical material is illustrated with numerical examples. In Section 4 "Results and Discussion," the ability of the proposed methods to solve the set tasks is experimentally demonstrated using benchmark datasets. Section 4 also contains limitations of the proposed methods. Lastly, Section 5 presents the main conclusions of the study and suggests potential areas for future research.

2. Basic Definitions and Concepts

Let us consider the feature matrix A of dimension $m \times k$, where m is the number of vectors obtained from the training set, k is the dimensionality (number of features) of the first feature set:

$$\underbrace{\begin{pmatrix} a_1^1 & a_2^1 & \cdots & a_k^1 \\ a_1^2 & a_2^2 & \cdots & a_k^2 \\ \cdots & \cdots & \cdots & \cdots \\ a_1^m & a_2^m & \cdots & a_k^m \end{pmatrix}}_{k} \Bigg\} m = A, \tag{1}$$

and the matrix B (obtained from the same training set) of dimension $m \times l$, where l is the dimensionality (number of features) of another feature space:

$$\underbrace{\begin{pmatrix} b_1^1 & b_2^1 & \cdots & b_l^1 \\ b_1^2 & b_2^2 & \cdots & b_l^2 \\ \cdots & \cdots & \cdots & \cdots \\ b_1^m & b_2^m & \cdots & b_l^m \end{pmatrix}}_{l} \Bigg\} m = B. \tag{2}$$

In general, k may equal, be less than, or be greater than l.

In practical tasks modeled in this way (there are two representations for the same objects at different sets of features), it is often necessary to express the feature vectors of one multidimensional space through their corresponding vectors from another feature space. That is, it is proposed to find such a matrix T that the following equation is satisfied:

$$B = TA, \tag{3}$$

where T is the transition matrix between matrices A (1) and B (2).

Note that in linear algebra, Formula (3) is a usual change of basis in vector space and, when the condition $m = k = l$ is met, finding T is trivial, i.e.,

$$T = BA^{-1}. \tag{4}$$

For the case being considered, where $m \neq k \neq l$, the inverse matrix does not exist, and thus, it is proposed to apply the generalization of the inverse matrix—the pseudoinverse matrix [27]. That is, it is proposed to find such a matrix T (4) of dimension $k \times l$, that ensures the transition between matrices A (1) and B (2):

$$AT \approx B. \tag{5}$$

Note that the approximation in Formula (5) is established concerning the Euclidean norm in the feature space of matrices.

Then, it is proposed to find T in the following way:

$$T \approx A^{+}B, \tag{6}$$

where A^{+} is the pseudoinverse matrix [27].

For the special case where $(A^{T}A)$ is of full rank, meaning the columns of matrix A are linearly independent ($\text{rank}(A) = k$), then A^{+} is defined as follows:

$$A^{+} = \left(A^{T}A\right)^{-1}A^{T}, \tag{7}$$

otherwise, for the full rank of (AA^{T}), meaning if the rows of matrix A are linearly independent ($\text{rank}(A) = m$), then

$$A^{+} = A^{T}\left(AA^{T}\right)^{-1}, \tag{8}$$

In practice, if the specified conditions are met, the mentioned methods formalized with Formulas (7) and (8) yield the best results. To decide on the use of Formula (7), it is sufficient to check that $\det\left(A^{T}A\right) \neq 0$, or similarly for Formula (8)—$\det\left(AA^{T}\right) \neq 0$.

In cases where it is not possible to obtain A^{+} as outlined in Formulas (7) or (8), meaning $\det\left(A^{T}A\right) = 0$, $\det\left(AA^{T}\right) = 0$ or they are close to 0, it is proposed to determine A^{+} using SVD decomposition [28]:

$$A^{+} = V\Sigma^{+}U^{T}, \tag{9}$$

where $A = U\Sigma V^{T}$ is the singular decomposition of matrix A, matrix Σ^{+} is formed by transposing matrix Σ and replacing all its non-zero diagonal elements with their reciprocals:

$$\Sigma^{+} = \left[D^{+}\vdots 0\right],\ m \geq k;$$

$$\Sigma^{+} = \begin{bmatrix} D^{+} \\ \cdots \\ 0 \end{bmatrix},\ m < k;$$

$$D^{+} = \text{diag}\left(\sigma_1^{+}, \sigma_2^{+}, \sigma_3^{+}, \ldots, \sigma_t^{+}\right),\ \sigma_i^{+} = \begin{cases} \frac{1}{\sigma_i^{+}},\ \sigma_i > 0; \\ 0,\ \sigma_i = 0. \end{cases}$$

Note that besides Formulas (7)–(9), there are other methods for determining A^+ ($A = BC$ decomposition, QR decomposition, decomposition using minors, etc.), which can be used if necessary.

Therefore, for any feature row vector a_j^*, $j = \overline{1,k}$, obtained from the model defined by matrix A, the corresponding feature row vector b_i^*, $i = \overline{1,l}$, defined by the model with matrix B, is determined using the obtained transition matrix T as follows:

$$b_i^* = Ta_j^*,\ i = \overline{1,k},\ j = \overline{1,l}. \tag{10}$$

In the next section, another application of the transition matrix T is proposed, specifically for obtaining the coefficients of the hyperplane from points that belong to (or are close to) this hyperplane. Here, we present the prerequisites for this, namely defining the coefficients of the hyperplane.

In some VA tasks, particularly when applying the principle of HITL, there arises a need to determine the coefficients of the hyperplane from points that belong to or (more often) are near this hyperplane [29]. The determination of the hyperplane coefficients is proposed in a manner analogous to work [30].

For a set of points $x(j)$, $j = \overline{1,m}$, which are located on a corresponding hyperplane, to determine the coefficients w_i and b of the equation of this hyperplane, a system of linear algebraic equations is formed [31]:

$$\begin{cases} w_1x_1(1) + w_2x_2\ (1) + \ldots + w_mx_m\ (1) + b = 0; \\ w_1x_1(2) + w_2x_2\ (2) + \ldots + w_mx_m\ (2) + b = 0; \\ \ldots \\ w_1x_1(m) + w_2x_2(m) + \ldots + w_mx_m\ (m) + b = 0. \end{cases} \tag{11}$$

The given homogeneous system (11) has an infinite number of solutions, but they are all proportional, and one of these solutions is proposed to be obtained by the method described below.

The system (11) is presented in the form of the equation of a hyperplane [32], which passes through m points:

$$\det \begin{pmatrix} x_1 & x_2 & \cdots & x_m & 1 \\ x_1(1) & x_2(1) & \cdots & x_m(1) & 1 \\ x_1(2) & x_2(2) & \cdots & x_m(2) & 1 \\ \vdots & \vdots & \cdots & \vdots & 1 \\ x_1(m) & x_2(m) & \cdots & x_m(m) & 1 \end{pmatrix} = 0, \tag{12}$$

and Formula (12) is decomposed using the first row:

$$\overbrace{\left(-1^{1+1}\right)\det \begin{pmatrix} x_2(1) & x_3(1) & \cdots & x_m(1) & 1 \\ x_2(2) & x_3(2) & \cdots & x_m(2) & 1 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ x_2(m) & x_3(m) & \cdots & x_m(m) & 1 \end{pmatrix}}^{w_1} x_1 +$$

$$+\overbrace{\left(-1^{1+2}\right)\det \begin{pmatrix} x_1(1) & x_3(1) & \cdots & x_m(1) & 1 \\ x_1(2) & x_3(2) & \cdots & x_m(2) & 1 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ x_1(m) & x_3(m) & \cdots & x_m(m) & 1 \end{pmatrix}}^{w_2} x_2 + \cdots +$$

$$+\overbrace{\left(-1^{1+k}\right)\det\begin{pmatrix} x_1(1) & \cdots & x_{k-1}(1) & x_{k+1}(1) & \cdots & x_m(1) & 1 \\ x_1(2) & \cdots & x_{k-1}(2) & x_{k+1}(2) & \cdots & x_m(2) & 1 \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ x_1(m) & \cdots & x_{k-1}(m) & x_{k+1}(m) & \cdots & x_m(m) & 1 \end{pmatrix}}^{w_k} x_k + \cdots + \tag{13}$$

$$\overbrace{\left(-1^{1+m}\right)\det\begin{pmatrix} x_1(1) & x_2(1) & \cdots & x_{m-1}(1) & 1 \\ x_1(2) & x_2(2) & \cdots & x_{m-1}(2) & 1 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ x_1(m) & x_2(m) & \cdots & x_{m-1}(m) & 1 \end{pmatrix}}^{w_m} x_m +$$

$$+\overbrace{\left(-1^{2+m}\right)\det\begin{pmatrix} x_1(1) & x_2(1) & \cdots & x_m(1) \\ x_1(2) & x_2(2) & \cdots & x_m(2) \\ \vdots & \vdots & \cdots & \vdots \\ x_1(m) & x_2(m) & \cdots & x_m(m) \end{pmatrix}}^{b}.$$

Considering the characteristics of the data used, the method of determining the coefficients does not always yield a positive result ($\det(*) = 0$). In such cases, it is proposed to obtain the coefficients w_i and b by solving an optimization problem using an evolutionary algorithm:

$$\underset{w_i, b \in [-1;1]}{\operatorname{argmin}} \ (\mathbf{W}x - b) = 0. \tag{14}$$

3. Materials and Methods

3.1. Interpretation of Results Obtained by DL Models through ML Models

In this work, we propose a novel XAI approach to the field of DL in general and validate it through two DL models, namely, convolutional neural network (CNN) and robustly optimized bidirectional encoder representations from transformers (RoBERTa). The authors understand DL here as a subset of the ML field based on artificial neural networks with feature learning. We put in the adjective "deep" the possibility of using multiple layers in a deep neural network (DNN). Within the proposed approach, the input information is the penultimate layer of the multilayer neural network. Such a layer can be extracted from a trained model of any type of neural network, such as convolutions, transformers, recurrent networks, autoencoders, etc.

3.1.1. Problem Statement and Proposed Solution Method

The rapid development of DL models based on various architectures, achieving results with DNNs for the same tasks (datasets) that are better than those obtained with ML models, on one hand, and the understandable explainability of results by ML models for humans, has led to a situation where the task of explaining results obtained by DL through ML models becomes highly relevant.

Let us consider a class of tasks where the input data are images, signals, etc., and the output information is the classification of these data into classes. Let us also assume that solutions to these tasks exist both through DL and ML means (for the same datasets). That is, we apply two comparable models for the same dataset. Here, we also consider the DL model as an FM, and the ML model as a MM. The side-by-side layout presented in Figure 1 serves to compare the feature extraction methodologies of the DL and ML models in capturing the informative attributes of the images.

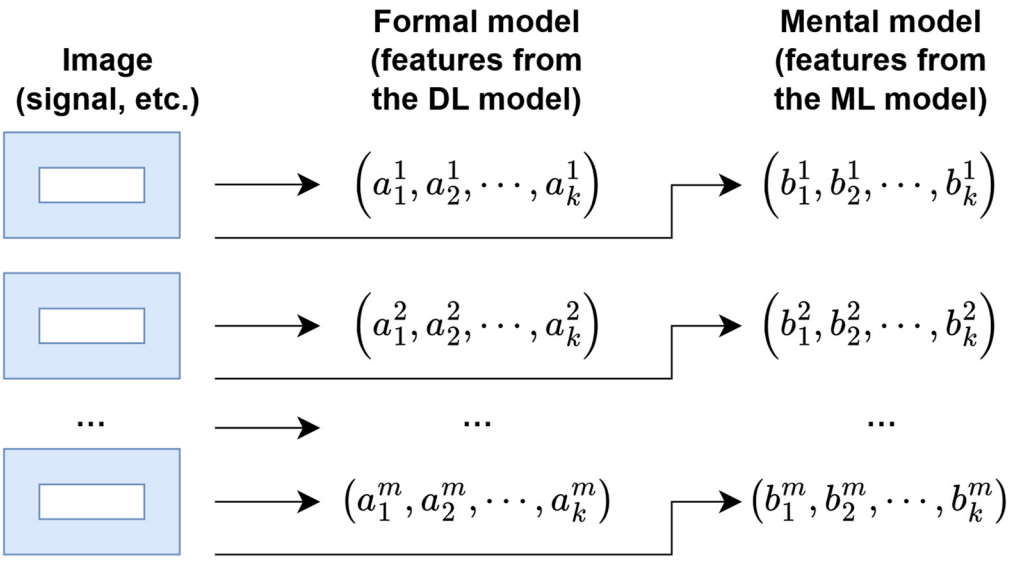


Figure 1. The process of feature extraction from a set of input data (first column) using two distinct computational models. The second column, labeled "Formal model", shows the feature vectors extracted by a DL model, denoted by a_i^j, where i indexes the feature and j—the image. The third column, labeled "Mental model", depicts the feature vectors extracted by an ML model, indicated by b_i^j. Each row corresponds to a single image processed by both models, highlighting the different dimensions k and l of the feature space represented by each model.

The approach described next correlates in a way with approximation, that is, describing one function (even if presented in a tabular form) in terms of another function (possibly also in tabular form). In this work, we propose to use a transition matrix between two models of characteristic features (presented as matrices) for the same set of input data as such a function and provide a mechanism for determining this transition matrix.

The proposed approach allows for mapping a vector composed of FM features (in the sense that the features are not understandable to an end user) into a new feature vector, whose features are obtained from MM ("white box" features that are understandable to an end user), serving as an interpretation (approximation) of the problem's solution. Here, MM is a space consisting of feature vectors—l-dimensional vectors of numerical features that represent a certain object. In our notation, FM is represented by matrix A (1), obtained from the DL model, and MM—by matrix B (2). The feature vector of FM, a k-dimensional vector of numerical features, could be, for example, a vector composed of the weight values of neurons from the penultimate layer of a DNN.

The task of explaining the results obtained by the DL model is proposed to be achieved through the transition matrix T (4) and transformation (10). This transition allows for obtaining results in MM features, which are understandable to an end user. The proposed method for obtaining the transition matrix T (4) is illustrated in Figure 2.

Below, we outline the main steps of the proposed method (Figure 2) for obtaining the transition matrix T (4).

Input Information: From the same dataset, corresponding to each annotated sample, matrix A is obtained as per structure (1) from the penultimate layer of a trained DNN, and matrix B is obtained as per structure (2) from a trained ML model. Note that each row of matrix A and the corresponding row of matrix B represents the same sample (object) from the training subset.

Step 1: Using any VA tool designed for dimensionality reduction of the feature space (for example, MDS), graphical representations of vectors as points on a plane [33,34] are obtained. It must be ensured that the mutual arrangement of these points for different models is similar (up to rotations), and the grouping of objects is formed according to the input annotated classes.

Step 2: The possibility of using Formulas (7), (8), or (9) for calculating A^+ is explored.

Step 3: The transition matrix T is calculated using Formula (6).

Step 4: Using the obtained matrix T, it is calculated according to Formula (10) for all vectors of the FM vector space. The vectors obtained by Formula (10) are compared with the vectors of the MM space to ensure the validity of the obtained transition matrix T.

Output Information: The transition matrix T.

After obtaining the transition matrix T following the steps outlined above and after obtaining the solutions from the trained DL model, we used Formula (10) to explain the results in the form of MM.

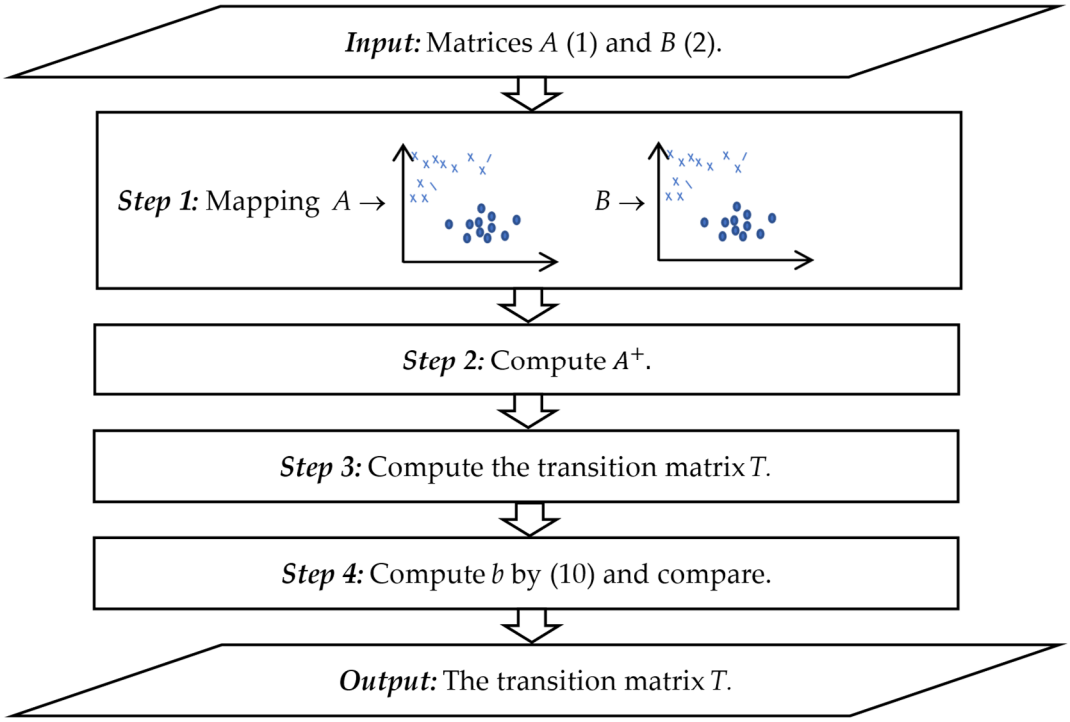


Figure 2. The scheme of the proposed method for transforming data from matrices A and B to the transition matrix T. The output of the method is the transition matrix T, which encapsulates the relationship between the two matrices.

Next, we provide a numerical example with synthetic data, illustrating the method steps. In the following Section 4, we show the results of experiments using the proposed approach based on benchmark datasets.

3.1.2. Illustrative Numerical Example

To illustrate the formalism proposed above, we took fifteen vectors $a_i^n, i = \overline{1,k}, n = \overline{1,15}$, $k = 5$, from the FM vector space (15). Note that these vectors belonged to three classes. Also, note that the features of these vectors are not understandable to humans:

$$\begin{array}{l}
\textit{Class 1}: \ a^1 = (2.8, 1.8, -2.8, 1.3, 0.4), \ a^2 = (2.9, -1.9, -2.9, 1.4, 0.5), \\
a^3 = (3, -2, -3, 1.5, 0.6), \ a^4 = (3.1, -2.1, -3.1, 1.6, 0.7), \ a^5 = (3.2, -2.2, -3.2, 1.7, 0.8); \\
\textit{Class 2}: \ a^6 = (-1.6, -2.5, 1.5, 0.2, 0.6), \ a^7 = (-1.3, -2.7, 1.3, 0.4, 0.8), \\
a^8 = (-1, -3, 1.5, 0.6, 1), \ a^9 = (-0.7, -3.2, 1.7, 0.8, 1.2), \ a^{10} = (-0.5, -3.5, 1.9, 1, 1.4); \\
\textit{Class 3}: \ a^{11} = (1.2, -1.2, 0.7, -0.3, -2.8), \ a^{12} = (1.1, -1.1, 0.8, -0.4, -2.9), \\
a^{13} = (1, -1, 0.844444, -0.44444, -3), \ a^{14} = (0.9, -0.9, 0.85, -0.45, -3.1), \\
a^{15} = (0.8, -0.8, 0.9, -0.5, -3.2).
\end{array} \tag{15}$$

It is noteworthy that vectors a^3, a^8, a^{13} were mutually perpendicular, i.e., $a^3 \perp a^8$, $a^3 \perp a^{13}$, $a^8 \perp a^{13}$. We took corresponding fifteen vectors (16) from the MM vector space (also note that the meaning of these features is understandable to humans):

$$\begin{array}{l} \textit{Class 1}: \ b^1 = (2.8, -1.8, -2.8, 1.3, 0.4), \ b^2 = (2.9, -1.9, -2.9, 1.4, 0.5), \\ b^3 = (3, -2, -3, 1.5, 0.6), \ b^4 = (3.1, -2.1, -3.1, 1.6, 0.7), \ b^5 = (3.2, -2.2, -3.2, 1.7, 0.8); \\ \textit{Class 2}: \ b^6 = (-1.6, -2.5, 1.5, 0.2, 0.6), \ b^7 = (-1.3, -2.7, 1.3, 0.4, 0.8), \\ b^8 = (-1, -3, 1.5, 0.6, 1), \ b^9 = (-0.7, -3.2, 1.7, 0.8, 1.2), \ b^{10} = (-0.5, -3.5, 1.9, 1, 1.4); \\ \textit{Class 3}: \ b^{11} = (1.2, -1.2, 0.7, -0.3, -2.8), \ b^{12} = (1.1, -1.1, 0.8, -0.4, -2.9), \\ b^{13} = (1, -1, 0.844444, -0.44444, -3), \ b^{14} = (0.9, -0.9, 0.85, -0.45, -3.1), \\ b^{15} = (0.8, -0.8, 0.9, -0.5, -3.2). \end{array} \tag{16}$$

Moreover, vectors b^3, b^8, b^{13} were also mutually perpendicular, i.e., $b^3 \perp b^8$, $b^3 \perp b^{13}$, $b^8 \perp b^{13}$.

Let us ensure that the mentioned vectors indeed form three classes. For this, we used a dimensionality reduction tool MDS, $R^5 \to R^2$ and $R^4 \to R^2$. The application results are shown in Figure 3 (vectors a^3, a^8, a^{13}, and b^3, b^8, b^{13}, are marked with a "+" sign).

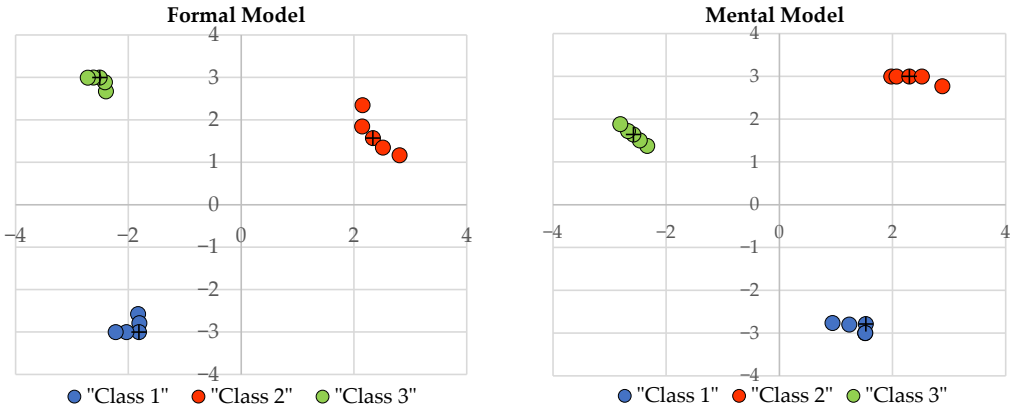


Figure 3. This figure contrasts the classification boundaries as determined by a Formal Model (FM) and a Mental (Human) Model (MM). Part (**a**) shows how the FM distinguishes three classes, evident by the clustering of class vectors in a two-dimensional feature space. Part (**b**) depicts the MM's classification, where similar clustering is observed, but with notable differences in the positioning and overlap of class vectors. Here, base vectors are marked with "+".

From Figure 3, it is evident that the vectors form three classes with quite wide inter-class gaps and, with accuracy to rotations of the coordinate grid, have a similar topology.

To start, we built matrices A and B from the three vectors a^3, a^8, a^{13}, and correspondingly, b^3, b^8, b^{13}:

$$A = \begin{pmatrix} 3 & -2 & -3 & 1.5 & 0.6 \\ -1 & -3 & 1.5 & 0.6 & 1 \\ 1 & -1 & 0.844444 & 0.44444 & -3 \end{pmatrix},$$

$$B = \begin{pmatrix} 1.843907868 & 1.998187 & -1.91286 & -1.97511 \\ 1.992023578 & -1.9238 & 0.706594 & -1.54378 \\ 1.107744254 & 1.615476 & 1.723582 & 1.807615 \end{pmatrix}.$$

We explored the possibility of using Formulas (7) or (8) to calculate A^+ by calculating $\det(A^T A)$. It equaled 0. Formula (7) was not suitable for the case under consideration (vectors in rows). Instead, Formula (9) was utilized for further calculations.

We determined the singular decomposition matrices $A = U\Sigma V^T$ as follows:

$$U = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \Sigma = \begin{pmatrix} 4.960846702 & 0 & 0 & 0 & 0 \\ 0 & 3.689173349 & 0 & 0 & 0 \\ 0 & 0 & 3.451176217 & 0 & 0 \end{pmatrix},$$

$$V = \begin{pmatrix} 0.604735478 & 0.271063435 & 0.289756285 & 0.57794 & -0.377950 \\ 0.403156985 & 0.813190305 & 0.289756285 & 0.272881 & 0.133279 \\ 0.604735478 & 0.406595152 & 0.2446833085 & 0.630521 & -0.107460 \\ 0.302367739 & 0.162638061 & 0.128780571 & 0.379846 & 0.849268 \\ 0.120947096 & 0.271063435 & 0.869268855 & 0.222893 & -0.326480 \end{pmatrix}.$$

Let us determine matrix Σ^+:

$$\Sigma^+ = \begin{pmatrix} 0.201578 & 0 & 0 \\ 0 & 0.271063 & 0 \\ 0 & 0 & 0.289756 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Let us determine A^+:

$$F^+ = V\Sigma^+U^T = \begin{pmatrix} 0.121901666 & 0.073475386 & 0.083958705 \\ 0.081267777 & 0.220426157 & 0.083958705 \\ 0.121901666 & 0.110213079 & 0.070898426 \\ 0.060950833 & 0.044085231 & -0.03731498 \\ 0.024380333 & 0.073475386 & 0.251876114 \end{pmatrix}.$$

Next, we calculated the transition matrix T using Formula (6):

$$T = A^+B = \begin{pmatrix} 0.278135369 & 0.520567817 & 0.140387778 & 0.024426 \\ 0.382248581 & 0.126035484 & 0.145008015 & 0.349038 \\ 0.522859856 & 0.341076002 & 0.433255464 & 0.198781 \\ 0.065904355 & 0.023301678 & 0.1497552201 & -0.25589 \\ 0.177604706 & -0.49953555 & 0.428847974 & -0.61688 \end{pmatrix}.$$

The calculations with the obtained matrix T using Formula (10) were conducted for all fifteen vectors of the FM vector space. We obtained fifteen predicted vectors of the MM space and compared them with the input values. The result is shown in Table 1.

Table 1. A comparison of predictions made by a Mental Model and those inferred by transforming a Formal Model's predictions using the transition matrix T to align with the Mental Model's framework, across fifteen instances.

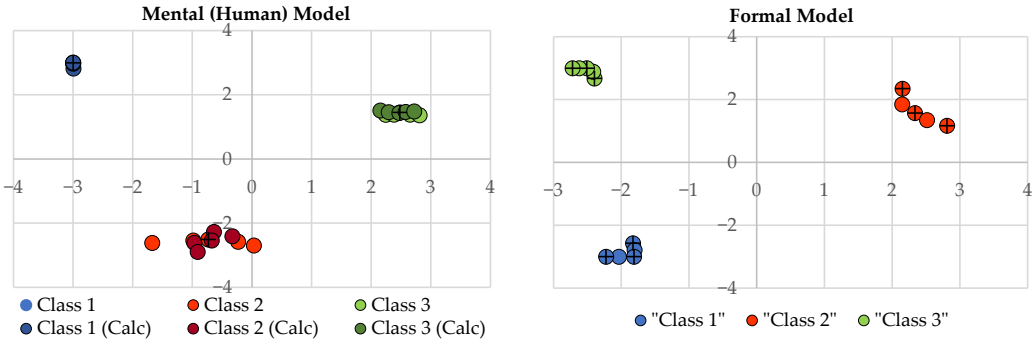
#	Mental Model				Formal Model → Mental Model				Diff
	b_1	b_2	b_3	b_4	b_1'	b_2'	b_3'	b_4'	
1	−1.97939	7.95931	−1.38111	−1.72964	−1.71146	1.95563	−1.71140	−1.69587	0.426
2	−1.97492	1.94851	−1.72661	−1.76121	−1.77768	1.97691	−1.81213	−1.83549	0.229
3	−1.84391	1.99818	−1.91285	−1.97511	−1.84391	1.99818	−1.91285	−1.97511	0
4	−1.99863	1.99967	−1.99844	−1.99976	−1.91013	2.01946	−2.01357	−2.11472	0.147
5	−1.99937	1.99889	−1.99960	−1.99891	−1.97636	2.04074	−2.11430	−2.25434	0.284
6	1.997776	−1.84400	1.660111	−1.37353	2.06518	−1.96399	0.94976	−1.03481	0.798
7	1.818753	−1.90968	1.20663	−1.40799	1.90491	−1.86938	0.73427	−1.31160	0.491
8	1.992024	−1.92380	0.70659	−1.54378	1.99202	−1.9238	0.70659	−1.54378	0
9	1.999174	−1.99759	0.21221	−1.58697	2.04090	−1.96562	0.66440	−1.74106	0.480
10	1.997854	−1.99941	−0.24340	−1.82758	2.15582	−2.0721	0.65076	−1.97568	0.922
11	0.851626	1.57420	1.58102	1.57393	1.00800	1.64037	1.55452	1.55364	0.173
12	1.008513	1.57079	1.59565	1.74176	1.07422	1.61909	1.65524	1.69325	0.112
13	1.107744	1.61547	1.72358	1.80761	1.10774	1.61547	1.72358	1.80761	0

Table 1. *Cont.*

#	Mental Model				Formal Model → Mental Model				Diff
	b_1	b_2	b_3	b_4	$b_1^{'}$	$b_2^{'}$	$b_3^{'}$	$b_4^{'}$	
14	1.089898	1.61136	1.88253	1.87352	1.11836	1.62421	1.76924	1.90428	0.121
15	1.290406	1.69528	1.95350	1.94625	1.15515	1.61882	1.84081	2.02117	0.206

The first half of Table 1 lists the predictions of vector values (b_1, b_2, b_3, b_4) by the MM. The second half shows the corresponding vector values (b_1', b_2', b_3', b_4') by the FM, translated into the MM's perspective. The final column "Diff" shows the distance between the corresponding vectors in Euclidean space (similarity), providing a clear metric for comparison. Table 1 illustrates the variance in outcome between the two models, which serves to understand the model agreement and the fidelity of translation between different modeling approaches.

We plotted the initial and calculated vectors on a graph (Figure 4) using MDS.



(**a**) (**b**)

Figure 4. Part (**a**) shows the Mental (Human) Model's classification points ("old") alongside the calculated points derived from the transition matrix T. Part (**b**) represents the Formal Model's ability to segregate three classes in the given feature space, mapping the underlying structure the model has learned; for each class, the number of base vectors was increased from one to three. Here, base vectors are marked with "+".

As can be seen from Figure 4a, the newly calculated vectors are in their respective classes. This scatter plot serves to illustrate the alignment between the human-generated classifications and the model-calculated estimations, thereby providing insights into the accuracy of the constructed transition matrix T in replicating human-like decision boundaries in a multidimensional feature space. Such an outcome suggests that even for a transition matrix composed of a few basis vectors, the result is satisfactory. Furthermore, we tried to improve it by increasing the number of basis vectors for calculation from one to three (Figure 4b)

Thus, matrix A and its corresponding matrix B take the following form:

$$A = \begin{pmatrix} 2.8 & -1.8 & -2.8 & 1.3 & 0.4 \\ 3.0 & -2.0 & -3.0 & 1.5 & 0.6 \\ 3.2 & -2.2 & -3.2 & 1.7 & 0.8 \\ -1.6 & -2.5 & 1.5 & 0.2 & 0.6 \\ -1.0 & -3.0 & 1.5 & 0.6 & 1.0 \\ -0.5 & -3.5 & 1.9 & 1.0 & 1.4 \\ 1.2 & -1.2 & 0.7 & -0.3 & -2.8 \\ 1.0 & -1.0 & 0.84 & -0.44 & -3.0 \\ 0.8 & -0.8 & 0.9 & -0.5 & -3.2 \end{pmatrix},$$

$$B = \begin{pmatrix} -1.97939 & 1.959307524 & -1.381119943 & -1.729640979 \\ -1.84391 & 1.99818664 & -1.912855282 & -1.97511053 \\ -1.99937 & 1.998896097 & -1.999605076 & -1.9989167665 \\ 1.997776 & -1.844000202 & 1.660111333 & -1.373532039 \\ 1.992024 & -1.923804827 & 0.706593926 & -1.543784398 \\ 1.997854 & -1.999410881 & -0.243400633 & -1.827587263 \\ 0.851626 & 1.574201387 & 1.581026838 & 1.573934081 \\ 1.107744 & 1.615475549 & 1.723582196 & 1.807614602 \\ 1.290406 & 1.695289797 & 1.953503509 & 1.946250271 \end{pmatrix}.$$

The matrices of the singular decomposition $F = U\Sigma V^T$, matrices Σ^+ and A^+ are provided in Appendix A.

Next, we calculated the transition matrix T using Formula (6):

$$T = \begin{pmatrix} -0.965871046 & 0.268676970 & -0.74950875 & 0.167864 \\ 0.040740026 & 0.228720439 & -0.724028187 & 0.576260 \\ 0.473706824 & -0.376589159 & -0.511934821 & 0.463701 \\ 1.978539601 & 0.622906549 & -1.45463279 & 0.357692 \\ -0.84067070 & -0.718752857 & -0.530542751 & -0.65073 \end{pmatrix}.$$

We conducted calculations with the obtained matrix T using Formula (10) for all fifteen vectors of the FM vector space. We obtained fifteen predicted vectors of the MM space and compared them with the input values. The result is presented in Table 2.

Table 2. The predicted vectors of a Mental Model as approximated by a Formal Model across fifteen instances. The table lists the component values of the vectors as predicted by the transition matrix T from the Formal Model to the Mental Model. The last column, "Diff", shows the distances in Euclidean space (similarity) between the vectors of the Mental Model and those predicted by the Formal Model.

#	Formal Model → Mental Model				Diff
	b_1	b_2	b_3	b_4	
1	−1.868316882	1.917325741	−1.465195993	−1.6609	0.160916565
2	−1.902561782	1.94939568	−1.615068122	−1.77742	0.133943398
3	−1.936806681	1.981465618	−1.76494025	−1.89393	0.19333596
4	−1.971051581	2.013535556	−1.914812378	−2.01044	0.089781324
5	−2.00529648	2.045605494	−2.064684507	−2.12696	0.151151106
6	2.045409347	−1.873238392	1.632130028	−1.33258	0.07472559
7	1.880332443	−1.782230819	1.257434896	−1.54882	0.206033576
8	1.900664267	−1.864730953	0.750368655	−1.6372	0.149932904
9	1.925070093	−1.924359043	0.170899595	−1.66796	0.138278882
10	2.041989021	−2.033726874	−0.261215771	−1.77314	0.08004815
11	0.88397757	1.609971462	1.532978489	1.549243	0.072418013
12	0.91822247	1.577901524	1.682850618	1.665757	0.146909333
13	1.03606919	1.601359125	1.780350637	1.776381	0.097646352
14	1.212437185	1.663686004	1.841190179	1.882882	0.139821386
15	1.321923723	1.681590851	1.943927409	1.994095	0.059680911

The "Diff" column in Table 2 contains the values of distances between the corresponding vectors in Euclidean space (similarity). As it is seen from the table, the obtained results were more promising compared to the previous ones. We obtained more similar vectors. In the next section, for real models, we calculated the matrix T using such a number of basis vectors from the training set to evenly cover the areas of the respective classes.

3.2. Obtaining a Separating Hyperplane Using the Human-in-the-Loop Principle

3.2.1. Problem Statement and Improved Method

VA is often used for data exploration and analysis in classification tasks. One of the VA tools for representing model objects in lower-dimensional spaces (to enable visualization) correlates with MDS. With such approaches for models represented as a set of n-dimensional feature vectors, corresponding feature vectors were obtained in Cartesian space. This allows for a visual assessment of the modeling quality, feature quality, and the model's ability to present classes as separate clusters. Essentially, we obtained a model in Cartesian space analogous to the input model.

Within the implementation of the HITL principle, having an analogous model in Cartesian space allows a person to indicate additional points belonging to lines that separate classes. This leads to the task of finding corresponding points in the input n-dimensional space. Typically, approaches analogous to the inverse MDS were used for this task.

The authors propose an improved method for obtaining object coordinates in n-dimensional space based on coordinates in Cartesian space. It involves using the transition matrix T with Formula (6). The method is outlined in Figure 5.

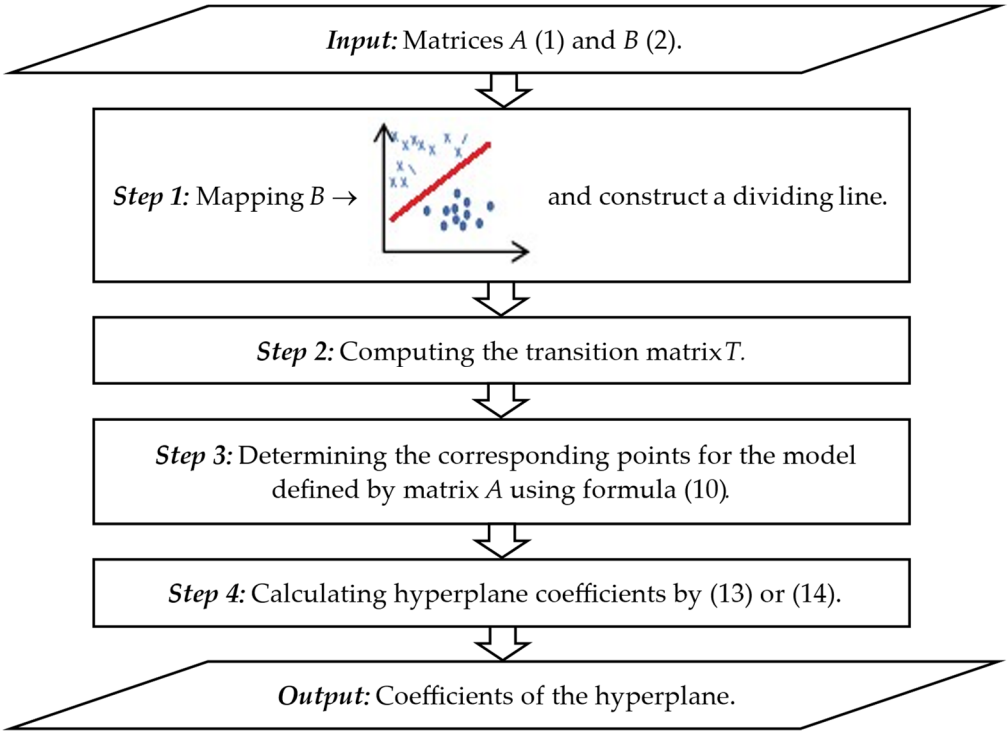


Figure 5. The scheme of the improved method for obtaining a separating hyperplane for classification tasks using a transition matrix based on VA and the HITL principle. The output of the method is the set of hyperplane coefficients calculated per Formulas (13) or (14), which form the delineation of decision boundaries in the feature space.

Below, we present the main steps of the improved method from Figure 5.

Input Information: From the input dataset, corresponding to each annotated sample, matrix A is obtained as per structure (1). Using dimensionality reduction tools (e.g., MDS), matrix B is received as per structure (2). Note that its dimensionality $l = 2$.

Step 1: The rows of matrix B are plotted on a graph. Visually, it is ensured that the groups of necessary vectors are separate. Empirically, the coordinates of two points are determined on the graph, based on which a line is drawn to separate the desired class (group of points on the graph) from other classes. Based on these two points, additional points are determined and located on the defined segment between the two found points.

The total number of points should equal the dimensionality of the input feature vector (matrix A). If we denote the found points as $c_1(x_1, x_2)$ and $c_k(x_1, x_2)$, then points c_i, $i = \overline{2, k-1}$, are defined as:

$$c_i = \frac{c_1 + (c_k - c_1)}{k-1},\ i = \overline{2, k-1}. \tag{17}$$

Step 2: For matrices A and B, the transition matrix T is calculated using Formula (6).

Step 3: For points c_i $i = \overline{1, k}$, using Formula (10), the corresponding points are determined for the model defined by matrix A (c_i^*, $i = \overline{1, k}$).

Step 4: Using points c_i^*, $i = \overline{1, k}$, and Formulas (13) or (14), the coefficients of the hyperplane are finally determined.

Output Information: Coefficients of the hyperplane, which can be used to determine the classes it separates.

Next, we provide a numerical example illustrating the steps of the method. Also, in the following Section 4, an experiment using the improved method with a benchmark dataset is described.

3.2.2. Illustrative Numerical Example

To illustrate the formalism proposed above, we took a set of vectors (15) from the previous example and formed matrix B. Then, using MDS, matrix A was derived from the data of matrix B as a reduction of the space dimensionality presented by matrix B to two-dimensional space:

$$B = \begin{pmatrix} 2.8 & -1.8 & -2.8 & 1.3 & 0.4 \\ 2.9 & -1.9 & -2.9 & 1.4 & 0.5 \\ 3.0 & -2.0 & -3.0 & -1.5 & 0.6 \\ 3.1 & -2.1 & -3.1 & 1.6 & 0.7 \\ 3.2 & -2.2 & -3.2 & 1.7 & 0.8 \\ -1.6 & -2.5 & 1.5 & 0.2 & 0.6 \\ -1.3 & -2.7 & 1.3 & 0.4 & 0.8 \\ -1.0 & -3.0 & 1.5 & 0.6 & 1.0 \\ -0.7 & -3.2 & 1.7 & 0.8 & 1.2 \\ -0.5 & -3.5 & 1.9 & 1.0 & 1.4 \\ 1.2 & -1.2 & 0.7 & -0.3 & -2.8 \\ 1.1 & -1.1 & 0.8 & -0.4 & -2.9 \\ 1.0 & -1.0 & 0.84 & -0.44 & -3.0 \\ 0.9 & -0.9 & 0.85 & -0.45 & -3.1 \\ 0.8 & -0.8 & 0.9 & -0.5 & -3.2 \end{pmatrix},\ A = \begin{pmatrix} -1.82751 & -2.5721 \\ -1.80564 & -2.78901 \\ -1.81244 & -2.99995 \\ -2.03578 & -2.99999 \\ -2.22591 & -2.99983 \\ 2.151466 & 2.349429 \\ 2.146416 & 1.848879 \\ 2.335564 & 1.57277 \\ 2.512781 & 1.35009 \\ 2.808853 & 1.168351 \\ -2.39704 & 2.674382 \\ -2.41546 & 2.889855 \\ -2.50885 & 2.999784 \\ -2.62157 & 2.999927 \\ -2.71964 & 2.998711 \end{pmatrix}.$$

We plotted the vectors from matrix A on a graph (Figure 6), and on this graph, we drew a segment separating Class 1 from Classes 2 and 3.

We empirically determined points $c_1(x_1, x_2)$ and $c_k(x_1, x_2)$, and calculated points c_i, $i = \overline{2, k-1}$, using Formula (17). Thus, we obtained the following coordinates of points that belonged to the line separating Class 1 from Classes 2 and 3: $c_1(2, -4)$, $c_2(0.5, -2.75)$, $c_3(-1, -1.5)$, $c_4(-2.5, -0.25)$, $c_5(-4, 1)$.

Next, using Formulas (6) and (9), similarly to the previous example, we determined the transition matrix T:

$$T = \begin{pmatrix} -0.62654 & -0.0371 & 0.388873 & 0.008528 & 0.622818 \\ -0.31919 & -0.09414 & 0.649254 & -0.23858 & -0.521214 \end{pmatrix}. \tag{18}$$

Using the obtained transition matrix T (18) and points c_i, $i = \overline{1, k}$, with Formula (10), we obtained the coordinates of corresponding points c_i^*, $i = \overline{1, k}$, that belonged to the hyperplane, as follows:

$$c_1^*(0.023677,\ 0.302352,\ -1.81927,\ 0.971392,\ 3.330592),$$
$$c_2^*(0.564504,\ 0.240333,\ -1.59101,\ 0.66037,\ 1.744817),$$
$$c_3^*(1.105331,\ 0.178314,\ -1.36275,\ 0.349349,\ 0.159041),$$
$$c_4^*(1.646158,\ 0.116295,\ -1.1345,\ 0.038327,\ -1.42673),$$
$$c_5^*(2.186986,\ 0.054277,\ -0.90624,\ -0.27269,\ -3.01251).$$

Then, using Formula (13) and an evolutionary algorithm with Formula (14), we determined the coefficients of the hyperplane that belonged to it or were nearby: $w_1 = -0.43893$, $w_2 = -0.47443$, $w_3 = -0.8868$, $w_4 = -0.61264$, $w_5 = -0.13863$, $b = -0.40266$.

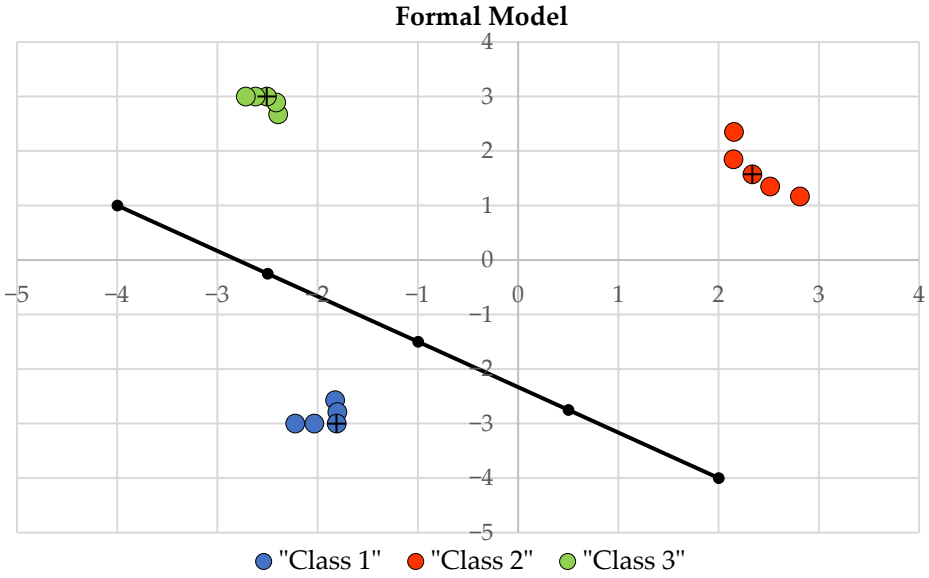


Figure 6. This figure illustrates the empirical derivation of a decision boundary within a Formal Model's feature space. The model has classified data points into three groups, represented as 'Class 1', 'Class 2', and 'Class 3' with different colors. Here, base vectors are marked with "+". A black line is drawn to demarcate the separation between 'Class 1' and the other classes, based on the model's classification criteria. This line exemplifies the model's decision-making process in differentiating between the classes, providing a visual understanding of the model's classification logic in a two-dimensional representation.

We validated the obtained hyperplane coefficients: substitute the vectors' coordinates (rows of matrix B) into the found hyperplane equation $\mathbf{W}x - b = 0$. Here, we ensured that the results of such substitution for vectors belonging to Class 1 had opposite values to the results for vectors belonging to Classes 2 and 3 (Table 3).

Table 3. This table exhibits the coefficients of a hyperplane defined by matrix B and their validation through a classification model. The coefficients b_1, b_2, b_3, b_4, and b_5 correspond to features of a multidimensional space, while the last column represents the evaluation of the hyperplane equation for each instance. Instances are categorized into three classes, with each class having distinct coefficient patterns.

Classes		The Model by Matrix B					$\mathbf{W}x - b = 0$
		b_1	b_2	b_3	b_4	b_5	
1	Class 1	2.8	−1.8	−2.8	1.3	0.4	0.853461
2		2.9	−1.9	−2.9	1.4	0.5	0.870563
3		3.0	−2	−3.0	1.5	0.6	0.887666
4		3.1	−2.1	−3.1	1.6	0.7	0.904769
5		3.2	−2.2	−3.2	1.7	0.8	0.921871

Table 3. *Cont.*

Classes		The Model by Matrix B					$\mathbf{W}x - b = 0$
		b_1	b_2	b_3	b_4	b_5	
6	Class 2	−1.6	−2.5	1.5	0.2	0.6	−0.05021
7		−1.3	−2.7	1.3	0.4	0.8	−0.0599
8		−1.0	−3.0	1.5	0.6	1.0	−0.37686
9		−0.7	−3.2	1.7	0.8	1.2	−0.74127
10		−0.5	−3.5	1.9	1.0	1.4	−1.01434
11	Class 3	1.2	−1.2	0.7	−0.3	−2.8	−0.40886
12		1.1	−1.1	0.8	−0.4	−2.9	−0.42597
13		1.0	−1.0	0.84	−0.44	−3.0	−0.42784
14		0.9	−0.9	0.85	−0.45	−3.1	−0.41905
15		0.8	−0.8	0.9	−0.5	−3.2	−0.42244

As seen from the $\mathbf{W}x - b = 0$ column in Table 3, the values for Class 1 vectors have an opposite sign to the values for Classes 2 and 3 vectors. Therefore, we have found a hyperplane that separates these classes.

3.3. Methodology and Experimental Setting

Computational experiments were conducted utilizing three renowned benchmark datasets, MNIST [35], FNC-1 [36], and Iris [37]. The selection of these datasets was motivated by their ability to facilitate the construction of relatively interpretable ML models. This interpretability is crucial for an objective evaluation of the proposed method's capacity to elucidate the outcomes generated by DL models.

3.3.1. Datasets

The Modified National Institute of Standards and Technology dataset (MNIST) is a large collection of handwritten digits widely used for training and testing in the field of ML and image processing. Each image in the MNIST dataset is 28 pixels in height and 28 pixels in width, for a total of 784 pixels. Each pixel has a single value associated with it, indicating the lightness or darkness of that pixel. These pixel values are integers between 0 and 255, inclusive. The digits are centered in a 28 × 28 image by computing the center of mass of the pixels and translating the image to position this point at the center of the 28 × 28 field. Each image is accompanied by a label indicating the actual digit it represents (from 0 to 9). MNIST provides a standard for researchers to compare the performance of their ML and DL models.

The Fake News Challenge Stage 1 (FNC-1) dataset is an essential benchmark for evaluating stance detection, a subtask of fake news detection. We refer to the problem statement of the stance detection task, which is defined in [38]. It contains 50,000 instances, each composed of a headline and a body text pair. Derived from the Emergent dataset [39], FNC-1 uses Emergent's 300 labeled claims as headlines and splits the associated articles into sentences to create body texts. The instances are labeled with four stances: "Agree", "Disagree", "Discuss", and "Unrelated", with a distribution of 7.4%, 2.0%, 17.7%, and 72.8%, respectively. Given its detailed annotations and the challenge of stance detection, FNC-1 serves as an excellent benchmark for assessing XAI approaches, as it requires models to provide interpretable reasoning for their stance predictions, aiding in the transparency and trustworthiness of fake news detection systems.

The Iris dataset, often referred to as Fisher's Iris dataset, is a classic dataset in the field of ML and statistics. The dataset is typically used for classification and pattern recognition in the context of statistical learning. There are 150 instances in the dataset, with 50 instances for each of the three Iris species. The dataset contains three classes, corresponding to each Iris species: Iris setosa, Iris versicolor, and Iris virginica. The four numerical features are sepal length, sepal width, petal length, and petal width. The Iris dataset allows for

easy visualization of the data and relationships between features, aiding in understanding feature importance and selection.

3.3.2. Design of Computational Experiments

In this work, we primarily evaluated the quality of the proposed approach, as assessing its effectiveness quantitatively is extremely challenging for the task of explanatory DL. The proposed approach, which involves constructing a transition matrix between the feature spaces of DL and ML models, as FMs and MMs, respectively, is a novel approach without obvious analogs.

Furthermore, we assume DL as a subset of the ML field based on artificial DNNs with feature learning. Within the proposed approach, the input information is the penultimate layer of the multilayer neural network. Such a layer can be extracted from a trained model of any type of DNN, such as transformers, recurrent networks, autoencoders, etc. Accordingly, the proposed contribution also relates to other DL models.

Below, we describe the methodology of our computational experiments to validate two proposed methods (see Figures 2 and 5).

To validate the proposed method based on a transition matrix, we constructed matrices B and A from the MNIST dataset in the following manner:

1. Construction of matrix B. For ML model (or MM) experiments, 10,000 vectors from the MNIST dataset training sample were randomly selected. These vectors were obtained from 28 × 28 MNIST images by "slicing" and concatenating the image rows. In other words, for matrix B, each 28 × 28 handwritten digit image was represented as a vector (one-dimensional array) with 784 features. Consequently, matrix B had a size of 10,000 × 784.
2. Construction of matrix A. A DL model (or FM) was created using a classic CNN architecture. The neural network was trained on the full MNIST dataset of 60,000 items. The architecture of the trained CNN is presented in Table A1 in Appendix B. From the penultimate layer of the created neural network model (490 weight coefficients), a vector was formed for each image from the formed sub-sample of 10,000, which was sequentially added to matrix A. As a result of these actions, two experimental matrices were obtained: matrix $A_{10,000\times490} = \mathbf{X}^{10,000}\mathbf{W}^{490}$ and matrix $B_{10,000\times784}$.
3. Calculation of the Transition Matrix T: Using Formula (6), the transition matrix T was calculated to find new representations for the rows of matrix A through basis B.

To further validate the proposed method based on a transition matrix for fake news detection, we constructed matrices B and A from the FNC-1 dataset as follows:

1. Construction of Matrix B: We randomly selected 10,000 labeled headline and article pairs from the training part of the FNC-1 dataset. Each article was preprocessed by tokenizing the text, removing stop words, and applying stemming. The processed text was then converted into a vector representation using the TF-IDF (Term Frequency-Inverse Document Frequency) encoding, resulting in one-dimensional vectors with a predefined number of features of 1024. As a result, we received matrix $B_{10,000\times1024}$, which served as an ML (or MM) model.
2. Construction of Matrix A: A RoBERTa model fine-tuned for fake news detection was crafted based on the architecture described in work [40]. The model was trained on the full FNC-1 dataset. The architecture of the trained RoBERTa model is presented in Table A2 in Appendix B. From the penultimate layer of the fine-tuned RoBERTa model, a vector of 768 was formed for each pair "headline/article" from the formed subsample of 10,000, sequentially added to matrix A. Consequently, two experimental matrices were obtained: matrix $A_{10,000\times768} = \mathbf{X}^{10,000}\mathbf{W}^{768}$ and matrix $B_{10,000\times784}$, where 768 is the size of the penultimate (fully-connected) layer of the constructed RoBERTa model.
3. Using Formula (6), the transition matrix T was calculated to find new representations for the rows of matrix A through basis B. The transition matrix T in this case was of 768 × 1024.

To validate an improved method for obtaining a separating hyperplane for classification tasks, we constructed matrices A and B based on the Iris dataset as follows:

1. Four numerical features, sepal length, sepal width, petal length, and petal width, formed matrix $B_{150\times4}$.
2. Dimensionality reduction of matrix $B_{150\times4}$ to a two-dimensional space was performed using MDS, resulting in matrix $A_{150\times2}$.
3. Matrix $A_{150\times2}$ was visualized on a graph; a separating line was also added to the graph to visually separate one class of the Iris dataset from all others.
4. Matrix $A_{150\times2}$ was used to construct the transition matrix T (6).
5. The transition matrix T was used to construct the separating hyperplane using Formula (13) and an evolutionary algorithm with Formula (14).

We propose our approach as a means of interpreting the DL model specifically for the end user, not the developer, as in the known approaches. In this regard, we conducted empirical validation based on a quantitative and qualitative comparison criterion (as presented below in Section 4), that is, how different the result obtained by the proposed method is from the results of the training and testing samples.

3.3.3. Quantitative Evaluation Criteria

To quantify the generated MNIST images within the experiments, we used the metrics Structural Similarity Index Measure (SSIM) and Peak Signal-Noise Ratio (PSNR) [41], providing a statistical analysis of the results.

The SSIM was used to assess the visual impact of three characteristics of an image: luminance, contrast, and structure, with values ranging between 0 and 1, where 1 indicates perfect similarity. The formula for SSIM is given by:

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)}, \tag{19}$$

where μ_x, μ_y are the average intensities of images x and y, σ_x^2, σ_y^2 are the variances, σ_{xy} is the covariance, and c_1, c_2 are constants.

PSNR, on the other hand, measures the ratio between the maximum possible power of a signal and the power of corrupting noise that affects its fidelity, typically expressed in a logarithmic decibel scale. The PSNR is calculated by:

$$\mathrm{PSNR} = 10\cdot \log_{10}\left(\frac{\mathrm{MAX}_I^2}{\mathrm{MSE}}\right), \tag{20}$$

where MAX_I is the maximum pixel value of the image, and MSE is the mean squared error between the original and the compressed image.

As for the FNC-1 dataset, given its imbalanced class distribution and the critical importance of distinguishing between "Agree", "Disagree", and "Discuss" for identifying fake news, the creators of FNC-1 recommended a weighted scoring approach [40]. Correctly identifying an unrelated instance adds 0.25 to the score. For related instances ("Agree", "Disagree", or "Discuss"), correctly labeling them as related in any way also adds 0.25. However, if the model accurately identifies the exact label of a related instance, it gains an additional 0.75. In summary, formulas for these weighted metrics are designed to reflect these considerations as follows.

The F_1-score is calculated using Precision and Recall of the test:

$$F_1 = 2\cdot\frac{\mathrm{Precision}\cdot\mathrm{Recall}}{\mathrm{Precision}\ +\ \mathrm{Recall}}, \tag{21}$$

where $\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$, $\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$, TP is true positive, FP represents false positive, and FN stands for false negative.

The macro-average F_1-score (F_1m) is the average of the F_1-scores for each class:

$$F_1m = \frac{F_{1\text{Agree}} + F_{1\text{Disagree}} + F_{1\text{Discuss}} + F_{1\text{Unrelated}}}{4}. \quad (22)$$

Weighted accuracy takes into account the imbalance of classes and is calculated as:

$$\begin{aligned}\text{Weighted accuracy} = 0.25 \cdot \text{Accuracy}_{\text{Related,Unrelated}} + \\ 0.75 \cdot \text{Accuracy}_{\text{Agree,Disagree,Discuss}}.\end{aligned} \quad (23)$$

4. Results and Discussion

This section includes a detailed description of computational experiments and their results. The section also discusses the proposed method for explaining results obtained by the DL model through the ML model and the improved method for constructing a separating hyperplane using the HITL principle. The limitations of the proposed methods are presented at the end of the section.

4.1. Results of Transitioning from the DL Model to the ML Model Using the Proposed Method

4.1.1. Visual Interpretation based on the MNIST Dataset

To demonstrate the ability of the proposed method to solve the set task, a subset of 10,000 training samples and a subset of 1000 testing samples were extracted from the MNIST dataset (see Section 3.3.2). The interpretation of the results was performed based on qualitative and quantitative analyses. The qualitative analysis involved a pairwise visual comparison of original MNIST handwritten digits with the handwritten digits generated using matrix T. The quantitative statistical analysis was conducted using SSIM (19) and PSNR (20) metrics, calculated for each pair of original MNIST vs. generated MNIST images.

A visual comparison between the original MNIST digits from the training dataset and the corresponding digits generated by our approach reveals both similarities and differences. For each digit in Figure 7, three images are presented: the original digit, the generated digit, and the difference image highlighting the discrepancies.

From Figure 7, it is evident that the overall structure and shape of the generated digits resemble the originals. This indicates that the transition matrix T accurately reproduces the main forms and sizes of MNIST digits. Moreover, the placement of digits within the image frame is consistent, indicating that the constructed matrix T aligns and centers the digits similarly to the original dataset.

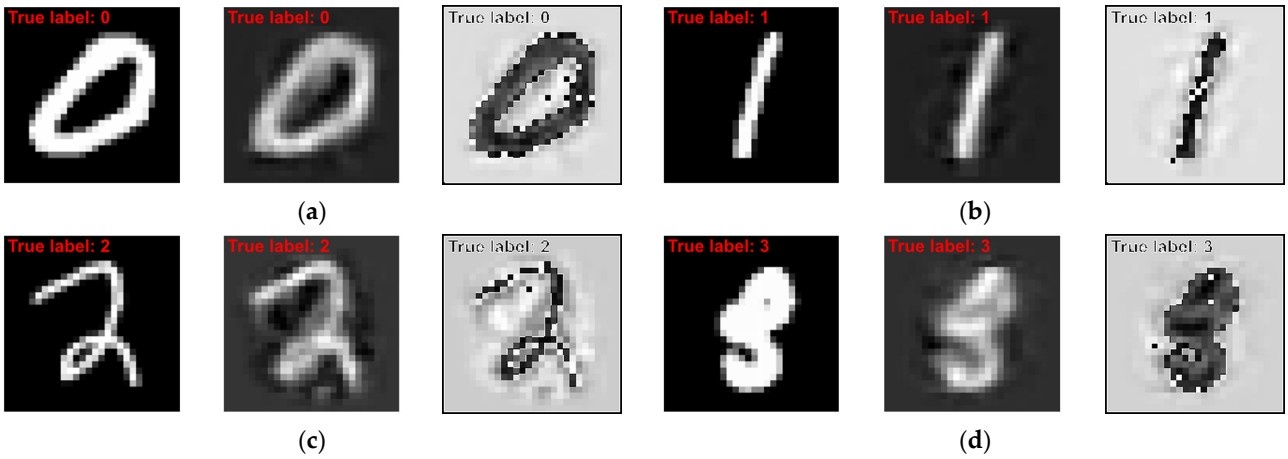


Figure 7. *Cont.*

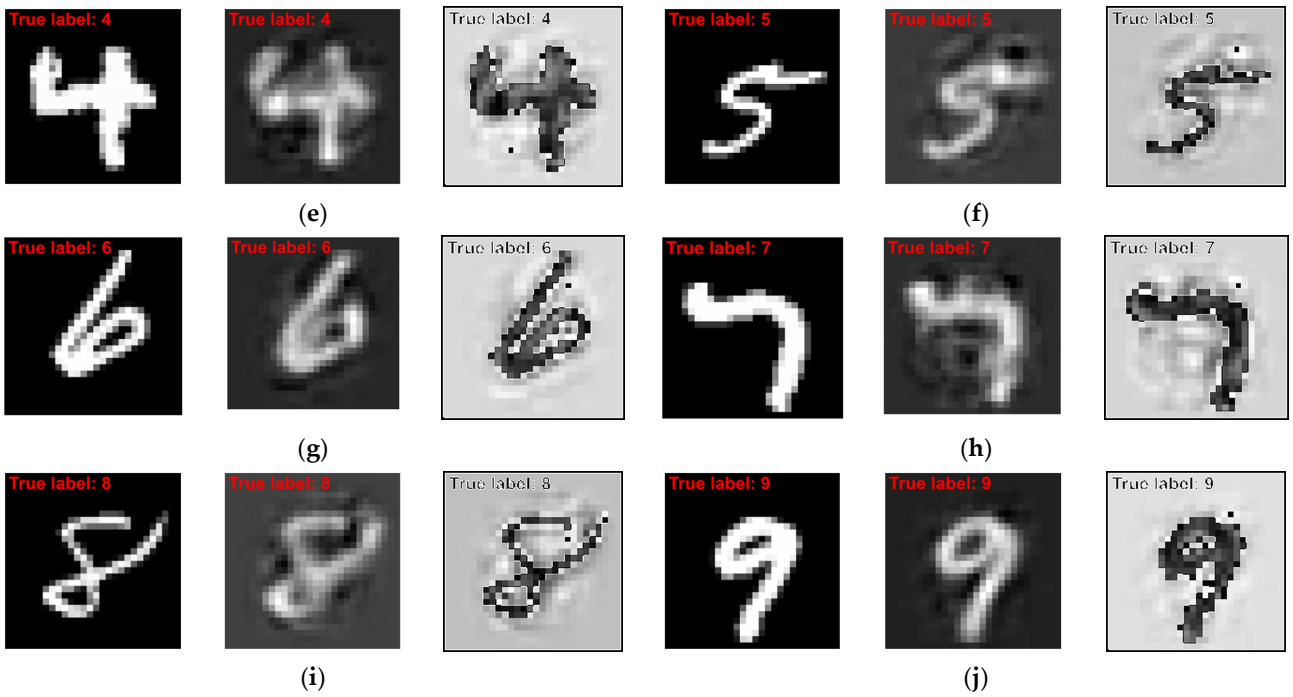


Figure 7. The original MNIST digits (left) extracted from the training dataset with the corresponding generated digits (center) using the transitioned matrix T and a difference image (right) highlighting the discrepancies between these digits: (**a**) zero, (**b**) one, (**c**) two, (**d**) three, (**e**) four, (**f**) five, (**g**) six, (**h**) seven, (**i**) eight, (**j**) nine.

Some of the generated digits have edges that are not as smooth and continuous as in the originals. This difference is particularly noticeable in curved digits, such as 0, 2, 3, etc. Some generated digits exhibit variations in line thickness, making certain parts of the digits appear bolder or fainter than the corresponding parts in the original images. These artifacts may be subtle but noticeable in different images.

Overall, like for the training MNIST dataset, the transition matrix T accurately reproduces the general appearance of MNIST digits and thus provides decent explainability for the CNN model. However, the presence of artifacts such as edge roughness, line thickness variations, and extraneous marks suggests the transitioning could be improved.

A visual analysis of original MNIST digits from the testing dataset and corresponding generated digits are presented in Figure 8.

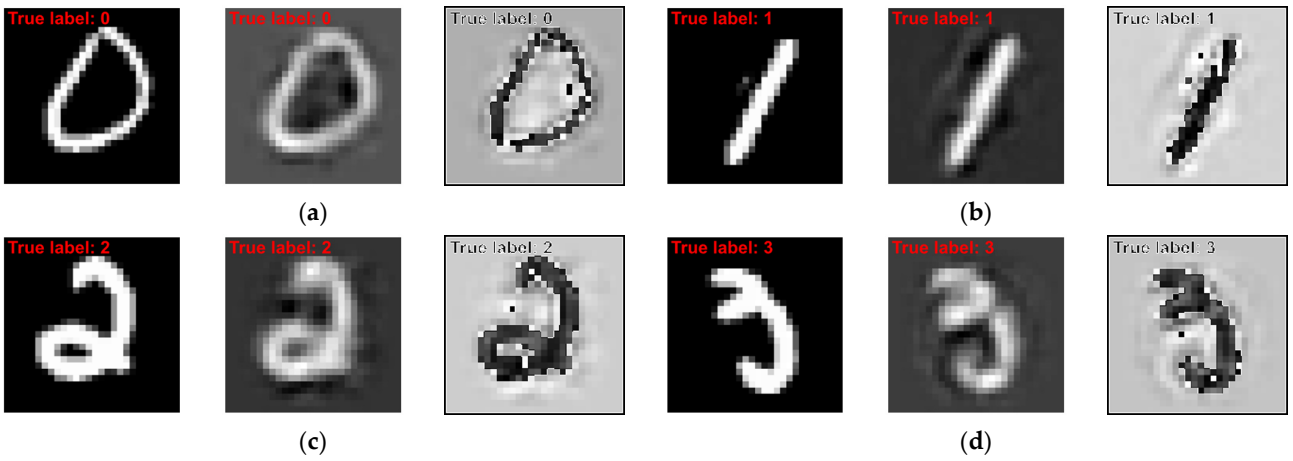


Figure 8. *Cont.*

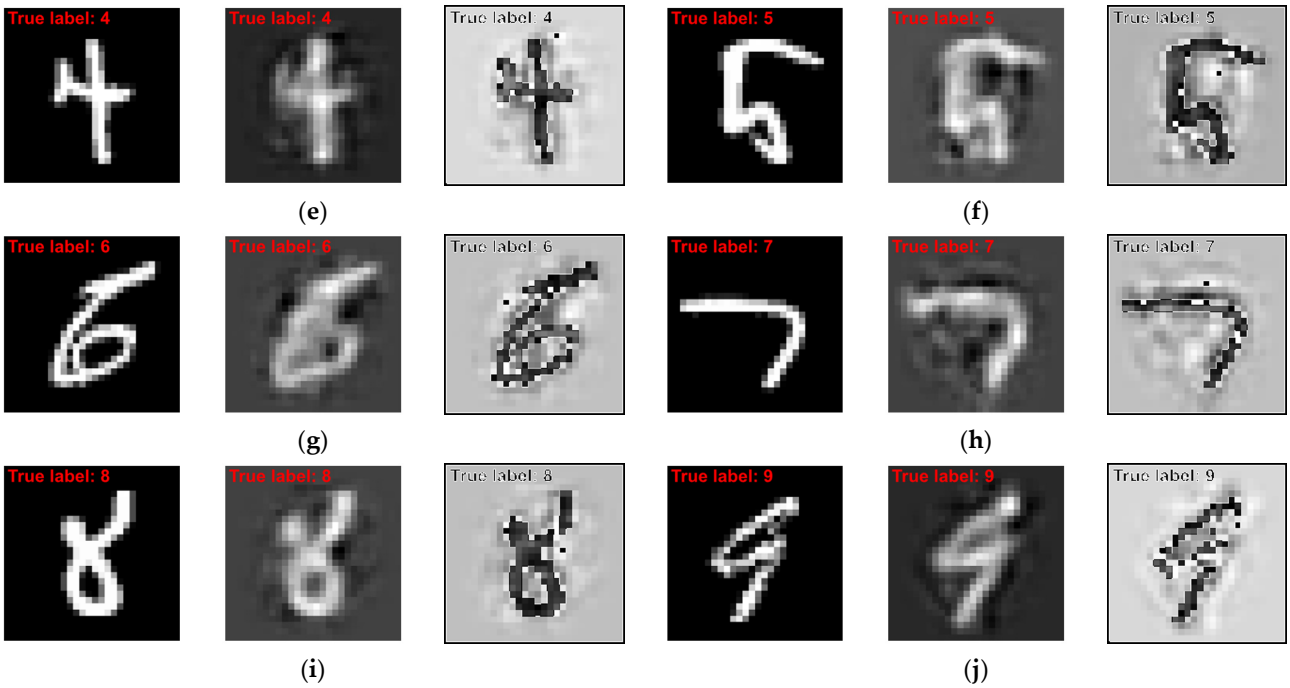


Figure 8. The original MNIST digits (left) extracted from the testing dataset with the corresponding generated digits (center) using the transitioned matrix T and a difference image (right) highlighting the discrepancies between these digits: (**a**) zero, (**b**) one, (**c**) two, (**d**) three, (**e**) four, (**f**) five, (**g**) six, (**h**) seven, (**i**) eight, (**j**) nine.

From Figure 8, it is clear that generated digits maintain the general shape and structure of the original digits. The size and proportions of the generated digits match the originals, indicating that the matrix T keeps the essential dimensions of each digit. Some generated digits from the testing dataset have a different inner texture or pattern, especially noticeable in digits like 2, 3, and 5, where significant discrepancies within the digit are visible in difference images (the third column). In some cases, the stroke thickness of generated digits differs from the original. In difference images, they stand out as isolated spots (Figure 8 (right)), especially in the background.

Consequently, based on the qualitative analysis, our approach allows for rigid replicating MNIST digits and gives a robust tool for explaining the outcome of the DL model.

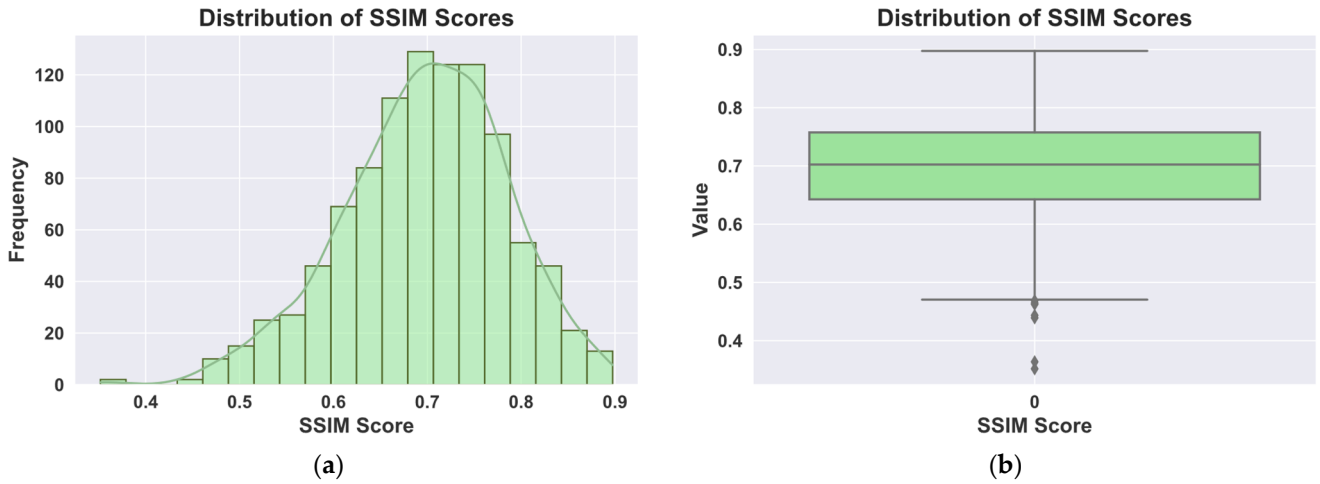
Next, we carried out the interpretation and implications of statistical indicators obtained from the original and generated MNIST images. The SSIM and PSNR scores for 1000 testing objects represent the degree of similarity of the generated MNIST handwritten digits to the originals and demonstrate statistics that are listed in Table 4.

Table 4. A statistical summary of image quality assessments using the SSIM and PSNR metrics based on the testing dataset. The table lists the mean, standard deviation, minimum, and specific percentiles (25th, 50th/median, 75th) along with the maximum values for both indicators.

Title 1	SSIM	PSNR
Mean	0.697	17.94
Standard Deviation	0.087	1.84
Minimum	0.352	11.46
25th Percentile	0.643	16.78
Median (50th Percentile)	0.702	17.76
75th Percentile	0.758	18.94
Maximum	0.898	24.08

From Table 4, we see that the average SSIM score of 0.697 indicates a rough similarity of the generated digits to the original MNIST digits. According to the SSIM scores, the generated digits are quite like the originals. The average and median PSNR values are relatively high, 17.94 and 17.76, respectively, indicating that on average the generated digits closely resemble the original MNIST digits. Overall, based on data in Table 4, the higher average and median indicate that the generated digits are of acceptable quality. The relatively low standard deviation suggests that our approach's execution is quite stable across different digits.

The distribution of SSIM scores, visualized through a histogram and a box plot in Figure 9, allows us to draw key conclusions.



(**a**) (**b**)

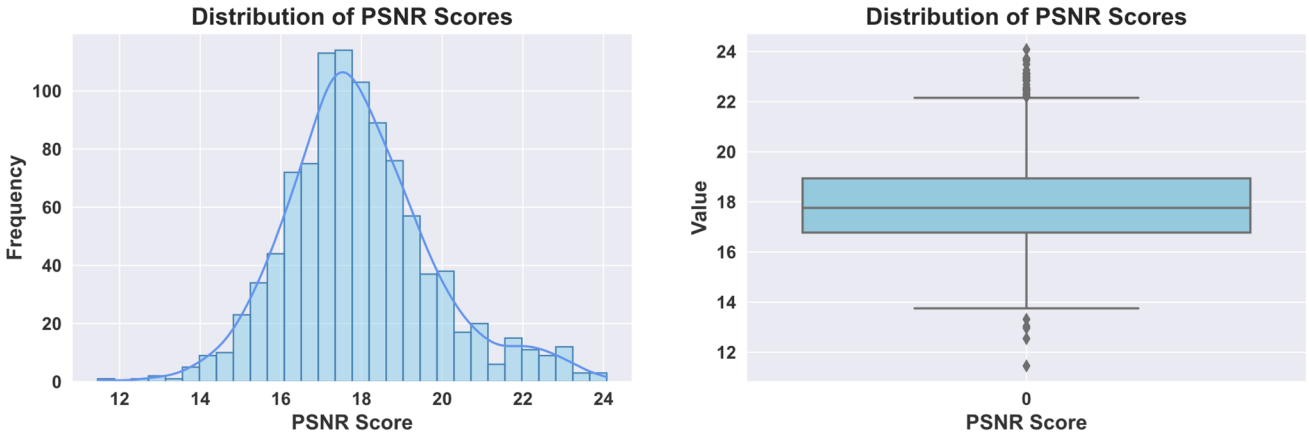
Figure 9. A visual representation of the distribution of SSIM scores across a dataset. Part (**a**) is a histogram displaying the frequency of SSIM scores, indicating the commonality of each score within the range observed. Part (**b**) is a box plot that summarizes the distribution of SSIM scores, highlighting the median, quartiles, and potential outliers.

The distribution in Figure 9 is roughly normal but slightly left-skewed (-0.388), indicating a concentration of scores around the mean and a tail of lower scores. This also points to several objects with lower similarity indicators. The kurtosis coefficient is about 0.186, suggesting a peak distribution close to normal. A significant number of scores cluster around the mean, indicating stable operation of the digit generation method for a considerable part of the dataset. Meanwhile, the range of scores from 0.35 to 0.90 indicates substantial variability in the results generated by the obtained transition matrix T.

The distribution of PSNR scores, presented in Figure 10 as a histogram and a box plot, provides valuable information on our approach's performance.

Specifically, the histogram in Figure 10a shows an approximately bell-shaped distribution, indicating that most PSNR scores are concentrated at the mean value, which is approximately 18. The histogram's peak around the mean value suggests that a significant portion of generated digits has PSNR values in this range, indicating stable model performance. There are no significant long tails on either side of the distribution that would indicate a significant number of extremely low or high scores.

Overall, from the values of SSIM and PSNR, we see that the proposed method with a transition matrix demonstrates an adequate level of reproducing original MNIST digits. Therefore, the conducted analysis confirms the functionality of our method in terms of explaining results obtained with a formal DL model through an ML model.



(**a**) (**b**)

Figure 10. A visual representation of the distribution of PSNR scores, which assesses image quality after compression or other processing. Part (**a**) presents a histogram detailing the frequency of various PSNR scores across a dataset, while part (**b**) shows a box plot summarizing the data's quartiles and highlighting outliers.

4.1.2. Interpreting Stance Detection Based on the FNC-1 Dataset

To see how well the transition matrix could explain the decisions made by the RoBERTa model when it comes to spotting fake news, we looked closely at the features after they were transformed and compared them to the original features based on TF-IDF. Our goal was to check if the transformed features could effectively link the headlines of articles to their main content. We conducted tests on detecting the stance of texts as follows.

First, we broke down the headlines and the main text into smaller pieces, changing the full text into a list of separate tokens (as detailed in Section 3.3.2). In this process, we removed common filler words and used stemming on the tokens to strip them down to their base forms. This step helps in treating similar words alike (for instance, changing "running" into "run") and makes the text data more uniform. Then, we transformed this cleaned-up text into the vector form using TF-IDF, which then became the input for our ML model (or MM) to produce matrix B.

We used a version of the RoBERTa model that was specifically adjusted for the FNC-1 dataset, as mentioned by [40], to tackle the task of detecting the stance. The penultimate (fully-connected) layer of this adjusted RoBERTa model gave us the feature vectors, forming FM space and thus creating matrix A.

Lastly, we constructed the transition matrix T using Formula (6). This matrix allowed us to change the feature vectors from FM space (the RoBERTa's penultimate layer) into MM space, meaning the TF-IDF features.

To evaluate the validity of our approach for the stance-detection task, we carried out computational experiments using the FNC-1 dataset and compared our findings with the leading approaches in identifying the stance of texts. This comparison included: (i) the original FNC-1 challenge baseline [42] that used feature engineering, (ii)–(iv) the top three systems from the FNC-1 challenge [43–45] as determined by their confusion matrices, (v) a recent study by Zhang et al. [46] that had notable success with the FNC-1 dataset, (vi) current state-of-the-art by Sepulveda-Torres et al. [40] in this task using the FNC-1 dataset, and (vii) our proposed approach that works by pulling out weights from the penultimate layer of the RoBERTa model and putting together the transition matrix T. The key metrics for comparison include the F_1-score (21) for four stance categories ("Agree", "Disagree", "Discuss", and "Unrelated") and two overall performance indicators: the macro-average F_1m (22) and weighted accuracy (23). The results shown in Table 5 are from applying all these approaches to the FNC-1 dataset.

Table 5. The proposed approach performance and its comparison with state-of-the-art systems over the testing split of the FNC-1 dataset. This table contains the performance for the class-wise F_1, macro-average F_1m, and weighted accuracy. The bolded numbers indicate the best-performing approach by the corresponding metric.

Approach	F_1-Score, %				F_1m, %	Weighted Accuracy, %
	Agree	Disagree	Discuss	Unrelated		
FNC-1 baseline [42]	45.62	12.46	62.32	92.40	51.60	74.67
Talos [43]	50.13	3.33	69.16	93.44	55.88	84.63
Athene [44]	45.29	14.21	70.98	93.62	57.98	85.01
UCLMR [45]	44.58	10.75	67.98	92.97	55.97	84.06
Zhang et al. [46]	62.75	**66.42**	**76.35**	92.98	77.68	89.40
Sepulveda-Torres et al. [40]	**69.02**	60.43	78.26	93.35	77.71	**89.42**
Our approach	68.96	60.44	78.22	**93.37**	**77.76**	89.38

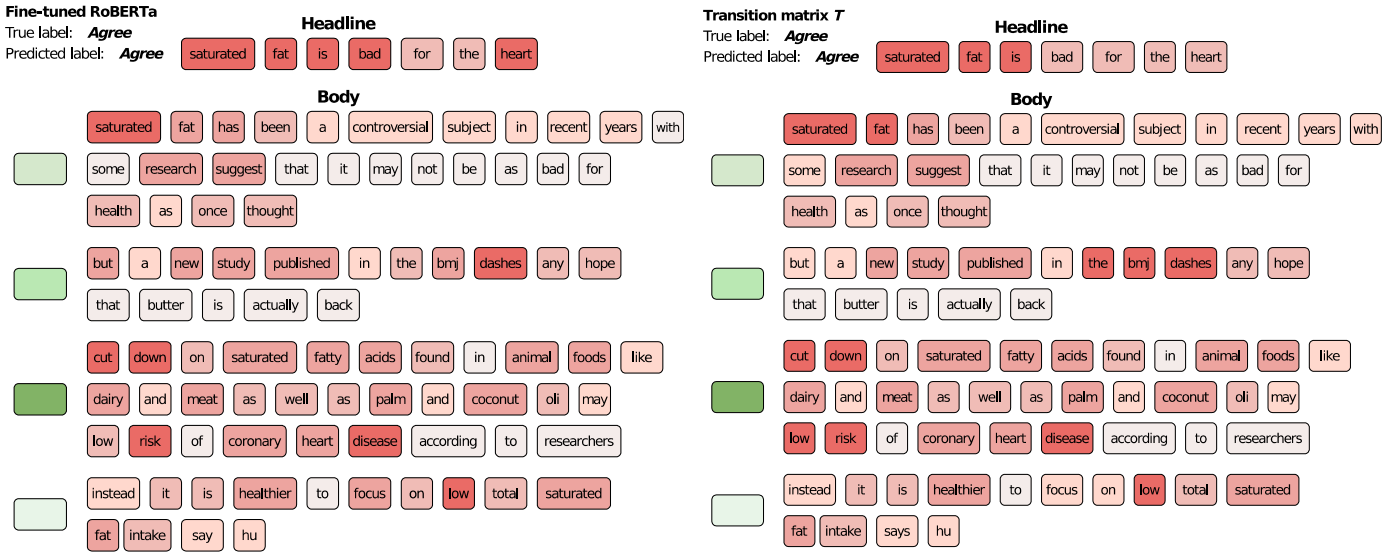
Looking at Table 5, our approach shows a strong F_1m score of 77.76%, which is just below the result of Sepulveda-Torres et al. [40] at 77.71%. This suggests that, on average, the performance of our transition model (from the RoBERTa to TF-IDF features) across all categories is consistent and robust compared to the fine-tuned RoBERTa model. The weighted accuracy of our approach is 89.38%, which is competitive and ranks third among all listed models. It slightly trails Sepulveda-Torres et al. [40] (89.42%) and Zhang et al. [46] (89.40%).

Overall, our approach based on the constructed transition matrix T demonstrates a balanced performance across various classes and overall metrics, suggesting it is a strong model for stance detection. Yet, there remains a slight scope for improvement in the "Agree" category and in maximizing the overall weighted accuracy. The results indicate a high performance in distinguishing "Disagree", which is often challenging due to the subtle cues that must be discerned, showcasing the effectiveness of our approach in handling complex linguistic patterns.

In addition to assessing the model's quality through its weighted accuracy as seen in Table 5, we also explored the possibility of gaining insights into the reasoning behind specific classification choices by examining the neural attention weights tied to both the sentence and the hierarchical encoders used for processing news article content. Our investigation into the model's logic involved looking at the transformed feature vector within the MM space to decode the steps leading to its final prediction. Key areas of focus included:

- Identifying crucial elements (such as specific words or phrases) within the transformed vector that played a significant role in the model's decision-making process.
- Matching these essential features against the original headline and article body to see if the model's attention was rightly placed on the main components of the text.
- Conducting a qualitative analysis of how interpretable the model's decisions were, based on the examination of the transformed features. We evaluated if the features identified as important within the MM space offered a clear and intuitive rationale for the model's predictions. For instance, a prediction of "Agree" supported by features that drew parallels between the headline and body text would signify that the model's reasoning was understandable and aligned with human logic.
- Checking for consistency in the model's focus across similar cases. If the model reliably pinpointed the same meaningful features as relevant in various instances of stance detection, it would indicate that our method, i.e., the use of the transition matrix T, ensured dependable interpretability.

Figure 11 shows a comparison between the attention weights calculated by the fine-tuned RoBERTa model (specifically from its penultimate layer) and those determined by our suggested transition matrix T for a single example taken from the FNC-1 testing dataset. Here, the darker shades highlight the words as red and sentences as green that the model deemed more important by assigning them higher weights.



(**a**) (**b**)

Figure 11. The comparison highlights the attention weights calculated using (**a**) the fine-tuned RoBERTa model and (**b**) our proposed transition matrix T for an instance example from the FNC-1 testing dataset. Here, darker shades indicate the words (in red) and sentences (in green) that were deemed more significant by the models, as shown by the heavier weights assigned to them.

The example presented in Figure 11, in which both the fine-tuned RoBERTa model and our approach using the proposed transition matrix T accurately classified an instance example into the "Agree" category, shows an interesting observation. In this illustration, it is evident that both approaches allocated similar levels of importance to certain parts of the text, as shown by the matching shades of red. Moreover, both the headline and the article's body include specific words like "saturated", "fat", "risk", and "disease" that strongly suggest a link between the headline's message and the main text, highlighting their relevance in determining the article's stance.

In sum, the obtained quantitative (Table 5) and qualitative (Figure 11) results with the FNC-1 dataset confirm the effectiveness of using the proposed transition matrix T for tasks involving the modeling/matching of long pieces of text. Based on the obtained results, we may conclude that the proposed approach can be a feasible solution to explaining the results obtained by DL models for the stance-detection task.

4.2. Results and Discussion for Obtaining a Separating Hyperplane Using the Human-in-the-Loop Principle

The ability of the improved method to construct a separating hyperplane was demonstrated using the Iris dataset as a reference. This was achieved using the VA method based on the HITL principle.

As per the computational methodology described in Section 3.3.2, matrices A and B for this task were constructed in the following way:

1. Four numerical features, i.e., sepal length, sepal width, petal length, and petal width, formed matrix $B_{150\times4}$.
2. Dimensionality reduction of matrix $B_{150\times4}$ to a two-dimensional space was performed using MDS, resulting in matrix $A_{150\times2}$.
3. Matrix $A_{150\times2}$ was visualized on a graph. Also, following the HITL principle, a separating line was added to the graph, visually separating one class of the Iris dataset from all others. The visualization result is demonstrated in Figure 12.

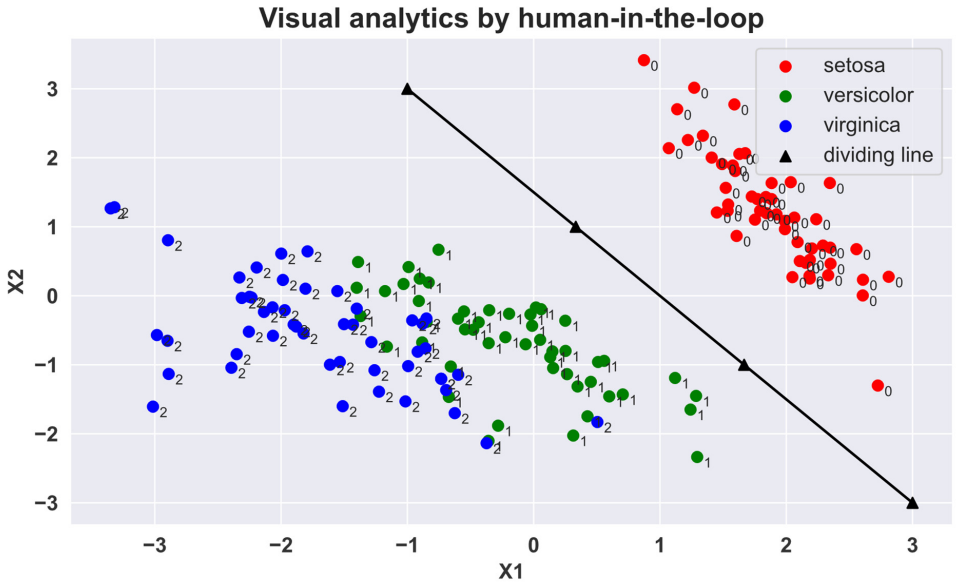


Figure 12. A scatter plot is used for visual analytics, with human-in-the-loop interaction facilitating the classification of three different species of the Iris flower. The data points are color-coded to represent "setosa" (red, label—"0"), "versicolor" (green, label—"1"), and "virginica" (blue, label—"2"), plotted against two feature axes, X1 and X2. A manually drawn dividing line (black) is included to show the decision boundary as determined by human analysis, which separates the "setosa" species from the others.

In Figure 12, we observe a clear separation of class 0 "setosa" from the other two classes—class 1 "versicolor" and class 2 "virginica."

4. Matrix $A_{150\times2}$ was used to construct the transition matrix T (6) using the approach proposed in this work.
5. Finally, the transition matrix T was used to construct the separating hyperplane using Formula (13) and an evolutionary algorithm with Formula (14). The coefficients of the hyperplane are as follows: $w_1 = 0.5 \times 10^{-5}$, $w_2 = 3 \times 10^{-5}$, $w_3 = -4.3 \times 10^{-5}$, $w_4 = -1.8 \times 10^{-5}$, $b = -3.1 \times 10^{-5}$.

Validation of the obtained hyperplane coefficients was performed similarly to the synthetic example (see Section 3.2.2): the coordinates of vectors (rows of matrix B) were substituted into the found hyperplane equation $\mathbf{W}x - b = 0$. The results of such substitution for vectors belonging to class 0 have opposite values to the results of substitution for vectors belonging to classes 1 and 2 (Table 6).

Table 6. This table details the validation of the linear classifier's hyperplane coefficients derived from the Iris dataset. It lists the sepal length, sepal width, petal length, and petal width for 15 out of 150 samples of three Iris classes—"setosa" (Class 0), "versicolor" (Class 1), and "virginica" (Class 2)—against the model's predictions. The last column evaluates whether each sample shown in the table belongs to one class (sign "+") or all other classes (sign "−").

Classes		The Model by Matrix B				$\mathbf{W}x - b = 0$
		Sepal Length	Sepal Width	Petal Length	Petal Width	
1	Class 0	5.1	3.5	1.4	0.2	0.000036713
2		4.9	3.0	1.4	0.2	0.000020684
3		4.7	3.2	1.3	0.2	0.000029910
4		4.6	3.1	1.5	0.2	0.000017786
5		5.0	3.6	1.4	0.2	0.000039177
...		...	...	...	...	...

Table 6. *Cont.*

Classes		The Model by Matrix B				$\mathbf{W}x - b = 0$
		Sepal Length	**Sepal Width**	**Petal Length**	**Petal Width**	
51	Class 1	7.0	3.2	4.7	1.4	−0.000125944
52		6.4	3.2	4.5	1.5	−0.000122347
53		6.9	3.1	4.9	1.5	−0.000139889
54		5.5	2.3	4.0	1.3	−0.000128922
55		6.5	2.8	4.6	1.5	−0.000138092
...		...	...	...	...	...
101	Class 2	6.3	3.3	6.0	2.5	−0.000202593
102		5.8	2.7	5.1	1.9	−0.000173582
103		7.1	3.0	5.9	2.1	−0.000195748
104		6.3	2.9	5.6	1.8	−0.000184621
105		6.5	3.0	5.8	2.2	−0.000196451
...		...	...	...	...	...

As seen from the $\mathbf{W}x - b = 0$ column in Table 6, the values for class 0 vectors have an opposite sign ("+") compared to the values for classes 1 and 2 vectors ("−"). As a result, the constructed hyperplane correctly separates these classes in the Iris dataset.

4.3. Limitations of the Proposed Methods

The proposed approach has certain limitations that are worth noting. First, our approach relies heavily on the quality of the input DL models. For it to be effective in particular cases, there needs to be a good degree of separability between the classes in the model's learned feature representations. If the classes are not well separated, even applying dimensionality reduction for visualization becomes very challenging. Since our methods are part of VA techniques, their usefulness depends on the model being able to accurately represent the data with distinct class separations.

Another issue is that to explain a DL model, we need an interpretable ML model to map the deep representations to. Thus, the interpretability is restricted by the characteristics of the interpretable model used. If this model is oversimplified compared to a corresponding DNN, we may lose fidelity in the explanations.

Next, for the method concerning the derivation of hyperplane coefficients, the model must have a class configuration that allows for feasible separation by a hyperplane or a set of hyperplanes. However, such a configuration is not universally attainable across all spatial arrangements of class distributions.

Finally, the examples we used focused primarily on classification tasks, which can be interpreted through other techniques like Class Activation Mapping (CAM) and Shapley additive explanations (SHAP) values [12,14,25]. A key advantage of our transition matrix approach is producing a global, model-level explanation spanning all classes, whereas methods like CAM and SHAP tend to be more localized. However, we agree that more validation is needed, including regression problems.

Overall, while using just MNIST, FNC-1, and Iris benchmarks limits the empirical validation, we aimed to first establish the core methodology. Going forward, we plan extensive experiments across diverse datasets and real-world sensitive scenarios, such as medical imaging, fake news detection, and financial forecasting, to comprehensively assess our method's effectiveness, robustness, and advantages over existing techniques.

5. Conclusions

In summary, the presented research addresses the challenge of enhancing the explainability of DL models. We proposed a novel approach that employs VA and a transition matrix methodology to improve the transparency and explainability of DL models. We validated our approach with experiments on the MNIST, FNC-1, and Iris datasets using a

qualitative and quantitative comparison criterion, that is, how different the results obtained by our approach are from the results of the training and testing samples. The quantitative results demonstrated that the proposed method based on the transition matrix T effectively replicated MNIST digits, achieving high similarity scores (SSIM: 0.697, PSNR: 17.94) and consistent performance with low standard deviations (SSIM: 0.087, PSNR: 1.84), proving to be a robust explanation tool for DL models. With an F_1m score of 77.76% and a weighted accuracy of 89.38%, our approach also showed strong performance in the stance-detection task based on the FNC-1 dataset. Furthermore, the qualitative results, particularly in recognizing critical words that bridge headlines and article bodies, complemented our quantitative achievements by illustrating our method's ability to capture nuanced textual relationships. In the classification task of the Iris dataset through a separating hyperplane, our improved method demonstrated enhanced accuracy and explainability, facilitating a deeper understanding of the model's decision-making process. Overall, our study marks a significant step towards making DL models more interpretable and accessible to end users, laying the groundwork for future advancements in XAI.

Our approach is limited by the quality of input DL models and the need for an interpretable ML model, along with the requirement for a class configuration that allows for hyperplane separation. To sum up, the transition matrix approach provides a global, model-level explanation, offering an advantage over more localized methods. Nonetheless, further validation, including regression problems, is necessary to fully assess its effectiveness.

Future work will explore the integration of the proposed approach with more complex DNN architectures, including deeper CNNs and transformers. Moreover, investigating the approach's application more thoroughly in real-world sensitive scenarios, such as medical imaging, fake news detection, and financial forecasting, will be crucial to understanding its practical implications and potential for societal impact.

Author Contributions: Conceptualization, O.B. and I.K.; methodology, O.B. and P.R.; software, P.R.; validation, P.R. and E.M.; formal analysis, O.B. and E.M.; investigation, O.B. and P.R.; resources, P.R.; data curation, P.R.; writing—original draft preparation, O.B. and P.R.; writing—review and editing, E.M.; visualization, P.R. and E.M.; supervision, I.K. and O.B.; project administration, I.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://github.com/radiukpavlo/transition-matrix-dl (accessed on 17 March 2024).

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

Abbreviations

The following abbreviations included in the text are reported alphabetically:

AI	Artificial Intelligence
CAM	Class Activation Mapping
CNN	Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Network
FACTS	Fairness, Accountability, Confidentiality, Transparency, and Safety
FM	Formal Model
FNC	Fake News Challenge
HITL	Human-in-the-Loop
MAX	Maximum Pixel Value
MDS	Multidimensional Scaling
ML	Machine Learning
MM	Mental Model
MNIST	Modified National Institute of Standards and Technology

MSE	Mean Squared Error
PSNR	Peak Signal-Noise Ratio
RoBERTa	Robustly Optimized Bidirectional Encoder Representations from Transformers
SHAP	SHapley Additive exPlanations
SSIM	Structural Similarity Index Measure
TF-IDF	Term Frequency-Inverse Document Frequency
VA	Visual Analytics
XAI	Explainable Artificial Intelligence

Appendix A

$$U = \begin{pmatrix} 0.52550 & 0.00993 & 0.02162 & -0.11637 & 0.33471 & -0.22843 & 0.71421 & 0.18061 & -0.05279 \\ 0.57238 & -0.00810 & -0.00094 & -0.06214 & -0.00136 & 0.30882 & -0.48291 & 0.57966 & -0.06210 \\ 0.61926 & -0.02613 & -0.02351 & -0.00791 & -0.33743 & -0.11937 & -0.15999 & -0.67604 & 0.06692 \\ -0.08930 & -0.49606 & 0.00709 & -0.71195 & 0.013792 & 0.05130 & -0.04140 & -0.10743 & -0.47215 \\ -0.00468 & -0.56303 & -0.00791 & -0.14997 & 0.087621 & -0.03326 & -0.02198 & 0.06644 & 0.80419 \\ 0.04501 & -0.65954 & -0.01190 & 0.65543 & -0.10516 & -0.02907 & 0.08224 & 0.06473 & -0.33223 \\ 0.04149 & -0.02699 & 0.55941 & 0.10925 & 0.43612 & 0.61605 & 0.04456 & -0.31745 & 0.01245 \\ 0.00042 & -0.00684 & 0.57862 & 0.02570 & 0.26943 & -0.66970 & -0.37025 & 0.04514 & -0.06538 \\ -0.03136 & 0.01669 & 0.59242 & -0.10475 & -0.70170 & 0.07481 & 0.28820 & 0.22664 & 0.06631 \end{pmatrix},$$

$$\Sigma = \begin{pmatrix} 8.66605 & 0 & 0 & 0 & 0 \\ 0 & 6.54804 & 0 & 0 & 0 \\ 0 & 0 & 5.96371 & 0 & 0 \\ 0 & 0 & 0 & 0.077947 & 0 \\ 0 & 0 & 0 & 0 & 0.09993 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$V = \begin{pmatrix} 0.61393 & 0.24125 & 0.28641 & 0.63737 & 0.27677 \\ -0.39214 & 0.81248 & -0.2790 & 0.27440 & -0.18152 \\ -0.60287 & -0.42332 & 0.23615 & 0.63369 & 0.00259 \\ 0.30254 & -0.17365 & -0.12562 & 0.22232 & -0.90172 \\ 0.11962 & -0.26890 & -0.87668 & 0.25974 & 0.27810 \end{pmatrix},$$

$$\Sigma^{+} = \begin{pmatrix} 0.115393 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.152717464 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.167681045 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.282922562 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10.00747 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$A^{+} = \begin{pmatrix} 0.87056 & -0.01436 & -0.89930 & -0.56822 & 0.09860 & 0.22300 & 1.32611 & 0.79484 & -2.00234 \\ -0.67256 & -0.04627 & 0.58002 & -0.33353 & -0.28124 & 0.33844 & -0.78519 & -0.50832 & 1.21357 \\ -0.12227 & -0.08988 & -0.05750 & -0.53988 & -0.08324 & 0.56916 & 0.12114 & 0.05121 & -0.07879 \\ -3.03603 & 0.01475 & 3.06553 & -0.31764 & -0.81853 & 1.15518 & -3.91402 & -2.43598 & 6.28816 \\ 0.89645 & -0.01611 & -0.92868 & -0.18077 & 0.21811 & -0.04479 & 1.16965 & 0.67365 & -2.07603 \end{pmatrix}.$$

Appendix B

Table A1. The architecture of the utilized CNN model for the MNIST visualization task. The table demonstrates the model's main layers and their corresponding hyperparameters.

Component	Layer Type	Input Channels	Output Channels	Kernel Size	Stride	Padding	Additional Details
conv_block_1	Conv2d	1	10	3×3	1	1	-
ReLU	-	-	-	-	-	-	
Conv2d	10	10	3×3	1	1	-	
ReLU	-	-	-	-	-	-	
MaxPool2d	-	-	2×2	2	0	-	
conv_block_2	Conv2d	10	10	3×3	1	1	-
ReLU	-	-	-	-	-	-	
Conv2d	10	10	3×3	1	1	-	
ReLU	-	-	-	-	-	-	
MaxPool2d	-	-	2×2	2	0	-	
Classifier Block	Flatten	-	-	-	-	-	start_dim = 1, end_dim = -1
Linear	-	-	-	-	-	in_features = 490, out_features = 10, bias = True	

Table A2. The architecture of the employed RoBERTa model for the stance detection task. The table shows the model's main encoder blocks and the corresponding number of features in each block.

Component	Layer Type	Input Features	Output Features	Additional Details
Embedding Block	Embedding Layer	Token IDs	768	Positional Embedding, Token Type Embedding
Encoder Block 1	Self-Attention	768	768	12 attention heads, Layer Normalization
	Feed-Forward	768	768	GELU Activation, Layer Normalization
Encoder Block 2	Self-Attention	768	768	12 attention heads, Layer Normalization
	Feed-Forward	768	768	GELU Activation, Layer Normalization
Encoder Block 3	Self-Attention	768	768	12 attention heads, Layer Normalization
	Feed-Forward	768	768	GELU Activation, Layer Normalization
...	...	...	...	...
Encoder Block 12	Self-Attention	768	768	12 attention heads, Layer Normalization
	Feed-Forward	768	768	GELU Activation, Layer Normalization
Classifier Head	Linear	768	Number of classes	Dropout, SoftMax Activation

Each encoder block in Table A2 consists of a self-attention layer and a feed-forward network, with layer normalization applied after each sublayer. This pattern repeats for each of the encoder blocks, though the specifics of each sublayer are consistent across blocks. Table A2 is truncated with "..." to indicate that blocks 4 through 11 are similar to blocks 1, 2, 3, and 12. The Classifier Head here is a linear layer that maps the output of the last encoder block to the number of classes for the stance detection task, with a dropout layer for regularization and a SoftMax layer for outputting probabilities.

References

1. Belle, V.; Papantonis, I. Principles and practice of explainable machine learning. *Front. Big Data* **2021**, *4*, 688969. [CrossRef] [PubMed]
2. Loyola-González, O. Black-box vs. White-box: Understanding their advantages and weaknesses from a practical point of view. *IEEE Access* **2019**, *7*, 154096–154113. [CrossRef]
3. Phillips, P.J.; Hahn, C.A.; Fontana, P.C.; Yates, A.N.; Greene, K.; Broniatowski, D.A.; Przybocki, M.A. *Four Principles of Explainable Artificial Intelligence*; NIST Interagency/Internal Report (NISTIR) 8312; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2021; p. 48. [CrossRef]
4. Casey, B.; Farhangi, A.; Vogl, R. Rethinking explainable machines: The GDPR's right to explanation debate and the rise of algorithmic audits in enterprise. *Berkeley Technol. Law J.* **2019**, *34*, 143–188. [CrossRef]
5. Greenstein, S. Preserving the rule of law in the era of artificial intelligence (AI). *Artif. Intell. Law* **2022**, *30*, 291–323. [CrossRef]
6. Laux, J.; Wachter, S.; Mittelstadt, B. Trustworthy artificial intelligence and the European Union AI act: On the conflation of trustworthiness and acceptability of risk. *Regul. Gov.* **2024**, *18*, 3–32. [CrossRef] [PubMed]
7. Chamberlain, J. The risk-based approach of the European Union's proposed artificial intelligence regulation: Some comments from a tort law perspective. *Eur. J. Risk Regul.* **2023**, *14*, 1–13. [CrossRef]
8. Maclure, J. AI, Explainability and public reason: The argument from the limitations of the human mind. *Minds Mach.* **2021**, *31*, 421–438. [CrossRef]
9. Merry, M.; Riddle, P.; Warren, J. A mental models approach for defining explainable artificial intelligence. *BMC Med. Inform. Decis. Mak.* **2021**, *21*, 344. [CrossRef] [PubMed]
10. Manziuk, E.; Barmak, O.; Krak, I.; Mazurets, O.; Skrypnyk, T. Formal model of trustworthy artificial intelligence based on standardization. In Proceedings of the 2nd International Workshop on Intelligent Information Technologies & Systems of Information Security (IntelITSIS-2022), Khmelnytskyi, Ukraine, 24–26 March 2021; Hovorushchenko, T., Savenko, O., Popov, P., Lysenko, S., Eds.; CEUR-WS: Aachen, Germany, 2021; Volume 2853, pp. 190–197.
11. Hoffman, R.R.; Mueller, S.T.; Klein, G.; Litman, J. Measures for explainable AI: Explanation goodness, user satisfaction, mental models, curiosity, trust, and human-AI performance. *Front. Comput. Sci.* **2023**, *5*, 1096257. [CrossRef]
12. Watson, D.S. Conceptual challenges for interpretable machine learning. *Synthese* **2022**, *200*, 65. [CrossRef]
13. Guidotti, R.; Monreale, A.; Pedreschi, D.; Giannotti, F. Principles of explainable artificial intelligence. In *Explainable AI Within the Digital Transformation and Cyber Physical Systems: XAI Methods and Applications*; Sayed-Mouchaweh, M., Ed.; Springer International Publishing: Cham, Switzerland, 2021; pp. 9–31. [CrossRef]
14. Rudin, C.; Chen, C.; Chen, Z.; Huang, H.; Semenova, L.; Zhong, C. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Stat. Surv.* **2022**, *16*, 1–85. [CrossRef]
15. Barredo Arrieta, A.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; Garcia, S.; Gil-Lopez, S.; Molina, D.; Benjamins, R.; et al. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **2020**, *58*, 82–115. [CrossRef]
16. Ras, G.; Xie, N.; van Gerven, M.; Doran, D. Explainable deep learning: A field guide for the uninitiated. *J. Artif. Intell. Res.* **2022**, *73*, 329–396. [CrossRef]
17. Izonin, I.; Tkachenko, R.; Kryvinska, N.; Tkachenko, P.; Greguš ml., M. Multiple linear regression based on coefficients identification using non-iterative SGTM neural-like structure. In Proceedings of the 15th International Work-Conference on Artificial Neural Networks (IWANN-2019), Gran Canaria, Spain, 12–14 June 2019; Rojas, I., Joya, G., Catala, A., Eds.; Springer International Publishing: Cham, Switzerland, 2019; Volume 11506, pp. 467–479. [CrossRef]
18. Fauvel, K.; Lin, T.; Masson, V.; Fromont, É.; Termier, A. XCM: An explainable convolutional neural network for multivariate time series classification. *Mathematics* **2021**, *9*, 3137. [CrossRef]
19. Bai, X.; Wang, X.; Liu, X.; Liu, Q.; Song, J.; Sebe, N.; Kim, B. Explainable deep learning for efficient and robust pattern recognition: A survey of recent developments. *Pattern Recognit.* **2021**, *120*, 108102. [CrossRef]
20. Olteanu, A.; Garcia-Gathright, J.; de Rijke, M.; Ekstrand, M.D.; Roegiest, A.; Lipani, A.; Beutel, A.; Olteanu, A.; Lucic, A.; Stoica, A.-A.; et al. FACTS-IR: Fairness, accountability, confidentiality, transparency, and safety in information retrieval. *SIGIR Forum* **2021**, *53*, 20–43. [CrossRef]
21. Choo, J.; Liu, S. Visual analytics for explainable deep learning. *IEEE Comput. Graph. Appl.* **2018**, *38*, 84–92. [CrossRef] [PubMed]
22. La Rosa, B.; Blasilli, G.; Bourqui, R.; Auber, D.; Santucci, G.; Capobianco, R.; Bertini, E.; Giot, R.; Angelini, M. State of the art of visual analytics for explainable deep learning. *Comput. Graph. Forum* **2023**, *42*, 319–355. [CrossRef]
23. Krak, I.V.; Kudin, G.I.; Kulyas, A.I. Multidimensional scaling by means of pseudoinverse operations. *Cybern. Syst. Anal.* **2019**, *55*, 22–29. [CrossRef]
24. Mosqueira-Rey, E.; Hernández-Pereira, E.; Alonso-Ríos, D.; Bobes-Bascarán, J.; Fernández-Leal, Á. Human-in-the-loop machine learning: A state of the art. *Artif. Intell. Rev.* **2023**, *56*, 3005–3054. [CrossRef]
25. La Barbera, D.; Roitero, K.; Mizzaro, S. Combining human intelligence and machine learning for fact-checking: Towards a hybrid human-in-the-loop framework. *Intell. Artif.* **2023**, *17*, 163–172. [CrossRef]
26. Retzlaff, C.O.; Das, S.; Wayllace, C.; Mousavi, P.; Afshari, M.; Yang, T.; Saranti, A.; Angerschmid, A.; Taylor, M.E.; Holzinger, A. Human-in-the-loop reinforcement learning: A survey and position on requirements, challenges, and opportunities. *J. Artif. Intell. Res.* **2024**, *79*, 359–415. [CrossRef]

27. Ben-Israel, A.; Greville, T.N.E. Existence and construction of generalized inverses. In *Generalized Inverses: Theory and Applications*; CMS Books in Mathematics; Springer: New York, NY, USA, 2003; pp. 40–51. [CrossRef]
28. Akritas, A.G.; Malaschonok, G.I. Applications of singular-value decomposition (SVD). *Math. Comput. Simul.* **2004**, *67*, 15–31. [CrossRef]
29. Kalyta, O.; Barmak, O.; Radiuk, P.; Krak, I. Facial emotion recognition for photo and video surveillance based on machine learning and visual analytics. *Appl. Sci.* **2023**, *13*, 9890. [CrossRef]
30. Krak, I.; Barmak, O.; Manziuk, E.; Kulias, A. Data classification based on the features reduction and piecewise linear separation. In Proceedings of the 2nd International Conference on Intelligent Computing and Optimization (ICO 2019), Koh Samui, Thailand, 3–4 October 2019; Vasant, P., Zelinka, I., Weber, G.-W., Eds.; Springer International Publishing: Cham, Switzerland, 2020; Volume 1072, pp. 282–289. [CrossRef]
31. Korn, G.A.; Korn, T.M. *Mathematical Handbook for Scientists and Engineers: Definitions, Theorems, and Formulas for Reference and Review*; Revised edition; Dover Publications: Mineola, NY, USA, 2000; p. 1152. ISBN 978-0-486-41147-7.
32. Kryvonos, I.G.; Krak, I.V.; Barmak, O.V.; Ternov, A.S.; Kuznetsov, V.O. Information technology for the analysis of mimic expressions of human emotional states. *Cybern. Syst. Anal.* **2015**, *51*, 25–33. [CrossRef]
33. Belkina, A.C.; Ciccolella, C.O.; Anno, R.; Halpert, R.; Spidlen, J.; Snyder-Cappione, J.E. Automated optimized parameters for T-distributed stochastic neighbor embedding improve visualization and analysis of large datasets. *Nat. Commun.* **2019**, *10*, 5415. [CrossRef] [PubMed]
34. Meilă, M.; Zhang, H. Manifold learning: What, how, and why. *Annu. Rev. Stat. Its Appl.* **2024**, *11*, 1–25. [CrossRef]
35. Deng, L. The MNIST database of handwritten digit images for machine learning research [Best of the Web]. *IEEE Signal Process. Mag.* **2012**, *29*, 141–142. [CrossRef]
36. Hanselowski, A.; PVS, A.; Schiller, B.; Caspelherr, F.; Chaudhuri, D.; Meyer, C.M.; Gurevych, I. A retrospective analysis of the fake news challenge stance-detection task. In Proceedings of the 27th International Conference on Computational Linguistics (COLING-2018), Santa Fe, NM, USA, 20–26 August 2018; Bender, E.M., Derczynski, L., Isabelle, P., Eds.; Association for Computational Linguistics: Santa Fe, NM, USA, 2018; pp. 1859–1874.
37. Unwin, A.; Kleinman, K. The Iris data set: In search of the source of Virginica. *Significance* **2021**, *18*, 26–29. [CrossRef]
38. Alturayeif, N.; Luqman, H.; Ahmed, M. A systematic review of machine learning techniques for stance detection and its applications. *Neural Comput. Appl.* **2023**, *35*, 5113–5144. [CrossRef] [PubMed]
39. Ferreira, W.; Vlachos, A. Emergent: A novel data-set for stance classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'2016), San Diego, CA, USA, 12–17 June 2016; Association for Computational Linguistics: Kerrville, TX, USA, 2016; pp. 1163–1168.
40. Sepúlveda-Torres, R.; Vicente, M.; Saquete, E.; Lloret, E.; Palomar, M. Exploring summarization to enhance headline stance detection. In Proceedings of the Natural Language Processing and Information Systems (NLDB-2021), Saarbrücken, Germany, 23–25 June 2021; Métais, E., Meziane, F., Horacek, H., Kapetanios, E., Eds.; Springer International Publishing: Cham, Switzerland, 2021; Volume 12801, pp. 243–254. [CrossRef]
41. Horé, A.; Ziou, D. Image quality metrics: PSNR vs. SSIM. In Proceedings of the 2010 20th International Conference on Pattern Recognition (ICPR-2010), Istanbul, Turkey, 23–26 August 2010; IEEE Inc.: New York, NY, USA, 2010; pp. 2366–2369. [CrossRef]
42. Galbraith, B.; Rao, D. FNC-1-Baseline: A Baseline Implementation for FNC-1. Available online: https://github.com/FakeNewsChallenge/fnc-1-baseline (accessed on 13 March 2024).
43. Baird, S.; Sibley, D.; Pan, Y. Talos Targets Disinformation with Fake News Challenge Victory. Available online: https://blog.talosintelligence.com/talos-fake-news-challenge/ (accessed on 13 March 2024).
44. Hanselowski, A. Description of the System Developed by Team Athene in the FNC-1. Available online: https://github.com/hanselowski/athene_system (accessed on 13 March 2024).
45. Riedel, B.; Augenstein, I.; Spithourakis, G.P.; Riedel, S. A simple but tough-to-beat baseline for the fake news challenge stance detection task. *arXiv* **2018**, arXiv:1707.03264. [CrossRef]
46. Zhang, Q.; Liang, S.; Lipani, A.; Ren, Z.; Yilmaz, E. From stances' imbalance to their hierarchical representation and detection. In Proceedings of the 2019 the World Wide Web Conference (WWW'19), San Francisco, CA, USA, 13–17 May 2019; Liu, L., White, R., Eds.; Association for Computing Machinery: New York, NY, USA, 2019; pp. 2323–2332. [CrossRef]