УДК 004.9

ИССЛЕДОВАНИЕ СЦЕНАРИЕВ ПРЕДВАРИТЕЛЬНОЙ ОБРАБОТКИ ДАННЫХ И ИНИЦИАЛИЗАЦИИ ВЛОЖЕНИЙ ПРИ РЕАЛИЗАЦИИ АЛГОРИТМА PACMAP

[1]Андрианова Е.Г., [1]Демидов Н.А.

[1]Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный университет информационных технологий, радиотехники и электроники» (МИРЭА), 119454, Россия, г. Москва, проспект Вернадского, 78, e-mail: andrianova@mirea.ru.

Цели. В статье рассмотрена задача нелинейного снижения размерности данных с применением алгоритма PaCMAP (Pairwise Controlled Manifold APproximation). Цель работы – исследование различных сценариев предварительной обработки данных и инициализации вложений при реализации алгоритма PaCMAP. **Методы.** В базовой версии алгоритма PaCMAP для предварительной обработки данных и инициализации вложений используется алгоритм PCA, являющийся алгоритмом линейного снижения размерности. В предлагаемой статье рассматриваются и исследуются в смысле минимизации значения функции потерь различные сценарии предварительной обработки данных и инициализации вложений с использованием 11 алгоритмов линейного и нелинейного снижения размерности в составе алгоритма PaCMAP. **Результаты.** Результаты экспериментальных исследований на тестовых и практических наборах данных демонстрируют преимущество ряда алгоритмов снижения размерности при включении их в сценарий предварительной обработки данных и инициализации вложений по отношению к алгоритму PCA. Лучшие результаты (в смысле минимизации функции потерь) на рассмотренных наборах данных получены, в частности, с применением алгоритмов UMAP, MSD и SE. При этом использование алгоритма MSD в составе алгоритма PaCMAP сопровождается значительными временными затратами. **Выводы.** Ряд алгоритмов линейного и нелинейного снижения размерности имеют преимущество (в смысле минимизации функции потерь) по отношению к алгоритму PCA при включении их в сценарии предварительной обработки данных и инициализации вложений в составе алгоритма PaCMAP. При этом использование алгоритма PCA в составе алгоритма PaCMAP обеспечивает минимальные временные затраты на его реализацию, а использование алгоритма MSD в составе алгоритма PaCMAP приводит к максимальным временным затратам на его реализацию.

Ключевые слова: алгоритм снижения размерности, PaCMAP, набор данных, визуализация, предварительная обработка данных, инициализация вложения, функция потерь.

SCENARIOS STUDY OF DATA PREPROCESSING AND EMBEDDING INITIALIZATION IN IMPLEMENTATION OF THE PACMAP ALGORITHM

[1]Andrianova E.G., [1]Demidov N.A.

[1]Federal State Budgetary Educational Institution of Higher Education "Moscow State University of Information Technologies, Radio Engineering and Electronics" (MIREA), 119454, Russia, Moscow, Vernadsky Avenue, 78, e-mail: andrianova@mirea.ru.

Objectives. This paper examines the problem of nonlinear data dimensionality reduction using the PaCMAP algorithm. The goal of the study is to explore the different scenarios of data preprocessing and embedding initialization when implementing the PaCMAP algorithm. **Methods.** The basic version of the PaCMAP algorithm uses the PCA algorithm, a linear dimensionality reduction algorithm, for data preprocessing and embedding initialization. This paper examines and explores various scenarios of data preprocessing and embedding initialization using 11 linear and nonlinear dimensionality reduction algorithms within the PaCMAP algorithm, in terms of loss function minimization. **Results.** Experimental studies on the test and real-world datasets demonstrate the advantages of several dimensionality reduction algorithms when included in the scenarios of data preprocessing and embedding initialization compared to the PCA algorithm. The best results (in terms of loss function minimization) on the examined datasets were obtained, in particular, using the UMAP, MSD, and SE algorithms. However, the use of the MSD algorithm within the PaCMAP algorithm is accompanied by significant time costs. **Conclusions.** A number of linear and nonlinear dimensionality reduction algorithms offer advantages (in terms of loss function minimization) over the PCA algorithm when included in the scenarios of data preprocessing and embedding initialization of the PaCMAP algorithm. Using the PCA algorithm within the PaCMAP algorithm minimizes its implementation time, while using the MSD algorithm within the PaCMAP algorithm results in the maximum implementation time.

Keywords: dimensionality reduction algorithm, PaCMAP, dataset, visualization, data preprocessing, embedding initialization, loss function.

Введение

В настоящее время существуют различные алгоритмы линейного и нелинейного снижения размерности данных, активно применяемые при решении практических задач. К алгоритмам линейного снижения размерности данных относятся такие алгоритмы, как FA (Factor Analysis) [1], LDA (Linear Discriminant Analysis) [2], PCA (Principal Component Analysis) [3]. К алгоритмам нелинейного снижения размерности данных относятся такие алгоритмы, как Isomap (Isometric Feature Mapping) [4], SLLE (Standard Locally Linear Embedding) [5], MLLE (Modified Locally Linear Embedding) [6], LLE с алгоритмом HE (Locally Linear Embedding with Hessian Eigenmap algorithm) [7], LLE с алгоритмом LTSA (LLE with Local Tangent Space Alignment algorithm) [8], MDS (Multidimensional Scaling) [9], Spectral Embedding (SE), называемый также Laplacian Eigenmaps [10], t-SNE (t-distributed Stochastic Neighbor Embedding) [11], LargeVis [12], UMAP (Uniform Manifold Approximation and Projection) [13], TriMAP (Triplets Manifold Approximation Projection) [14], PaCMAP (Pairwise Controlled Manifold Approximation Projection) [15] и другие.

Большинство этих алгоритмы таковы, что способны сохранять только глобальную или только локальную структуру многомерных данных при вложении их в пространство низкой размерности из исходного пространства высокой размерности.

Так, алгоритмы PCA [3] и MDS [9] хорошо сохраняют глобальную структуру многомерных данных при выполнении их вложения в пространство низкой размерности. Преобразования, лежащие в основе этих алгоритмов, не позволяют сохранить сложную нелинейную структуру многомерных данных при выполнении их вложения в пространство низкой размерности. Алгоритм TriMAP [14], предложенный в 2019 году, является одним из современных алгоритмов нелинейного снижения размерности, в основном реализующим сохранение глобальной структуры многомерных данных.

Алгоритмы Isomap [4], LLE [5 – 8], SE [10], а также – более современные алгоритмы, такие как t-SNE [12], LargeVis [13] и UMAP [14], предложенные соответственно в 2008, 2016 и 2018 годах, хорошо сохраняют локальную структуру многомерных данных при выполнении их вложения в пространство низкой размерности. Алгоритмы Isomap [4], LLE [5 – 8], SE [10] пытаются сохранять локальные евклидовы расстояния, вычисленные в исходном пространстве высокой размерности при создании вложения в пространство низкой размерности. Однако это не всегда приводит к сохранению структуры соседства, так как точки данных, которые являются соседями друг друга в пространстве высокой размерности, не всегда являются соседями в пространстве низкой размерности. Алгоритмы t-SNE [12], LargeVis [13] и UMAP [14] пытаются сохранять локальную структуру многомерных данных посредством сохранения структуры графа (то есть посредством сохранения для каждой точки данных её соседей).

Как утверждают авторы алгоритма PaCMAP [15], предложенного в 2019 году, он позволяет сохранять одновременно глобальную и локальную структуры многомерных данных при выполнении их вложения в пространство низкой размерности благодаря реализованной в нём передовой концепции. В частности, в алгоритме PacMAP вложение в пространство низкой размерности осуществляется с учетом информации о парах точек таких трёх типов, как пары близких точек (near pairs), пары средне-близких точек (mid-near pairs) и пары дальних точек (further pairs), а также с использованием функции потерь специального вида, подвергающейся трехэтапной минимизации с настраиваемыми поэтапно весовыми коэффициентами для слагаемых, соответствующих трем группам пар точек. При этом для инициализации вложения многомерных данных в пространство низкой размерности используется алгоритм PCA. Этот же алгоритм применяется в алгоритме PaCMAP к точкам данных перед построением графа k ближайших соседей. По мнению авторов алгоритма PaCMAP, это позволяет значительно ускорить алгоритм PaCMAP без существенной потери точности. Кроме того, авторы алгоритма PaCMAP обеспечивают в нём возможность случайной инициализации начального вложения с применением многомерного нормального распределения. При этом авторы алгоритма PaCMAP считают, что он достаточно (но не полностью) устойчив к выбору инициализации из-за своих пар средне-близких точек и динамической структуры графа.

Алгоритмы снижения размерности могут быть использованы не только для получения двухмерных или трёхмерных наборов данных с целью проведения их визуального анализа [16, 17], но и для получения наборов данных произвольной размерности, меньшей, чем размерность исходного пространства, с целью использования этих наборов, например, при разработке классификаторов данных [18, 19].

Целью настоящей работы является исследование различных сценариев предварительной обработки данных и инициализации вложений при реализации алгоритма PaCMAP в смысле минимизации значения функции потерь при приемлемых временных затратах. При этом исследование выполняется как на трёх известных тестовых наборах данных, так и на трёх практических наборах данных, сформированных при решении задач в сфере онлайн-обучения студентов Института информационных технологий (ИИТ) РТУ МИРЭА программированию на языке Python.

1. Аспекты реализации алгоритма PaCMAP

Алгоритм PaCMAP предполагает выполнение вложения многомерного набора данных, заданного в пространстве высокой размерности, в пространство низкой размерности с сохранением как локальной, так и глобальной структур данных.

Пусть X – набор данных, описывающий точки данных в пространстве высокой размерности. Пусть Y – набор данных, описывающий точки данных в пространстве низкой размерности. При этом точки данных из набора Y сопоставлены точкам данных из набора X.

Пусть n – общее число точек данных, а n_{NB} – число ближайших соседей, которое по умолчанию в алгоритме PaCMAP полагается равным 10. Пусть J – число итераций, в ходе которых выполняется оптимизация функции потерь алгоритма PaCMAP (по умолчанию $J = 450$).

Реализация алгоритма PaCMAP заключается в выполнении следующих шагов.

1. Найти множество пар точек данных разного типа.

1.1. Найти близкие пары точек данных: для каждой i-й точки ($i = \overline{1,n}$), вычислив n_{NB} ближайших соседей на основе масштабированных квадратов расстояний:

$$d_{ij}^2 = \frac{\|\mathrm{x}_i - \mathrm{x}_j\|^2}{\sigma_{ij}}, \quad (1)$$

где x_i, x_j – векторы i-й и j-й точек данных соответственно; $\sigma_{ij} = \sigma_i \cdot \sigma_j$; σ_i– среднее расстояние между i-й точкой данных и её ближайшими соседями с 4-го до 6-го; σ_j – среднее расстояние между j-й точкой данных и её ближайшими соседями с 4-го до 6-го.

Следует отметить, что сначала необходимо найти $min(n_{NB} + 50, n)$ ближайших соседей для i-й точки данных в соответствии с евклидовым расстоянием, а затем на основе этого подмножества соседей необходимо вычислить масштабированные расстояния d_{ij}^2 и выбрать n_{NB} ближайших соседей в соответствии с этими расстояниями.

1.2. Найти средне-близкие пары точек данных: для каждой i-й точки ($i = \overline{1,n}$) сначала случайным образом выбрать 6 точек данных, а затем из них выбрать вторую ближайшую точку и поставить её в пару с i-й точкой. Число таких пар определяется как $n_{MN} = \lfloor n_{NB} \cdot MN_{ratio} \rfloor$, где MN_{ratio} – отношение числа средне-близких пар к числу близких пар (по умолчанию $MN_{ratio} = 0.5$).

1.3. Найти дальние пары точек данных: для каждой i-й точки ($i = \overline{1,n}$) случайным образом выбрать точку, не являющуюся соседней для неё, и поставить её в пару с i-й точкой. Число таких пар определяется как $n_{FP} = \lfloor n_{NB} \cdot FP_{ratio} \rfloor$, где FP_{ratio} – отношение числа дальних пар к числу близких пар (по умолчанию $FP_{ratio} = 2$).

2. Выполнить инициализацию низкоразмерного вложения Y точек данных.

3. Минимизировать функцию потерь в течение J итераций в три этапа, используя алгоритм оптимизации Adam [20, 21]:

$$Loss = \omega_{NB} \cdot \sum_{i,j-\text{близкие точки}} \frac{\tilde{d}_{ij}}{10+\tilde{d}_{ij}} + \omega_{MN} \cdot \sum_{i,k-\text{средне-близкие точки}} \frac{\tilde{d}_{ik}}{10000+\tilde{d}_{ik}} + \omega_{FP} \cdot \sum_{i,l-\text{дальние точки}} \frac{\tilde{d}_{il}}{1+\tilde{d}_{il}}, \quad (2)$$

где $\tilde{d}_{ab} = \|\mathrm{y}_a - \mathrm{y}_b\|^2 + 1$.

На первом этапе (на первых 100 итерациях алгоритма оптимизации Adam):

$\omega_{NB} = 2$, $\omega_{MN}(t) = 1000 \cdot \left(1 - \frac{t-1}{100}\right) + 3 \cdot \frac{t-1}{100}$, $\omega_{FP} = 1$. $\omega_{MN}(t)$ убывает линейно от 1000 до 3.

На втором этапе (на 101 – 200 итерациях алгоритма оптимизации Adam):

$\omega_{NB} = 3$, $\omega_{MN} = 3$, $\omega_{FP} = 1$.

На третьем этапе (на 201 – 450 итерациях алгоритма оптимизации Adam):

$\omega_{NB} = 1$, $\omega_{MN} = 0$, $\omega_{FP} = 1$.

4. Получить низкоразмерное вложение Y точек данных.

Масштабированные расстояния d_{ij}^2 вычисляются только для выбора соседей: они позволяют учесть, что соседство (в смысле расстояния) в разных частях пространства признаков, описывающих точки данных, может быть измерено существенно разными величинами.

Так как число ближайших соседей n_{NB} обычно на порядок меньше общего числа точек данных [15], использование случайного выбора точек данных позволяет эффективно выбирать неближайших соседей при формировании средне-близких и дальних пар точек данных. При этом использование стратегии случайного выбора позволяет существенно снизить вычислительные затраты на формирование искомых пар.

Авторы алгоритма PaCMAP считают, что результаты его работы почти нечувствительны к алгоритму инициализации, и предлагают использовать на шаге 2 алгоритм PCA для минимизации времени работы алгоритма PaCMAP [15]. Альтернативно авторы предлагают использовать случайную инициализацию с использованием многомерного нормального распределения $N(0; 10^{-4} \cdot I)$, где I – единичная матрица [15]. Кроме того, авторы предлагают выполнять предобработку данных перед построением графа k ближайших соседей в алгоритме PaCMAP [15]. При этом они утверждают, что использование алгоритма PCA позволяет существенно ускорить работу алгоритма PaCMAP без значительной потери точности [15].

Минимизация функции потерь $Loss$ алгоритма PaCMAP выполняется на шаге 3 в три этапа. На первом этапе выполняется улучшение первоначального размещения встроенных точек до такого, которое сохраняет, в определенной степени, как глобальные, так и локальные структуры, но в основном глобальные структуры. Это достигается посредством сильного взвешивания средне-близких пар, при этом веса этих пар постепенно уменьшаются, что позволяет постепенно переориентироваться с учета глобальной структуры на учет локальной структуры. На втором этапе выполняется улучшение локальной структуры при сохранении глобальной структуры, захваченной на первом этапе посредством назначения небольшого (но не нулевого) веса для средне-близких пар. На этих двух этапах оптимизации алгоритм PaCMAP пытается избежать локальных оптимумов, используя процесс, который имеет сходство с имитацией отжига и техникой «раннего преувеличения» (early exaggeration), используемой t-SNE [11]. При этом алгоритм PaCMAP фокусируется сначала на средне-близких парах, а затем уже на соседях. На третьем этапе выполняется улучшение локальной структуры посредством снижения веса средне-близких пар до нуля, а веса близких пар до меньшего значения, чтобы подчеркнуть роль отталкивающей силы, помогающей разделить возможные кластеры и сделать их границу более четкой. Этот этап оказывает большее влияние на наборы данных с преимущественно локальной структурой по сравнению с наборами данных с глобальной структурой.

Следует отметить, что алгоритм PaCMAP стремится отдавать предпочтение глобальной структуре, когда такая структура существует, а также сохраняет локальную структуру, когда такая структура существует.

Рассматриваемые алгоритмы снижения размерности имеют отличающиеся друг от друга функции потерь. Поэтому сравнить результаты работы различных алгоритмов снижения размерности, ориентируясь только на значения функций потерь, невозможно. При этом во многих случаях функции потерь алгоритмов снижения размерности задаются сложными математическими формулами, в чём можно убедиться, проанализировав, например, различные варианты функции потерь для алгоритма UMAP [22, 23]. Достоинством функции потерь $Loss$ (2) алгоритма PaCMAP является простота её представления.

В ходе экспериментальных исследований используется программная реализация алгоритма PaCMAP, предложенная в [24].

2. Экспериментальные наборы данных

В качестве экспериментальных наборов данных рассматриваются три известных тестовых набора данных [25 – 27] и три практических набора данных, сформированных при решении задач в сфере онлайн-обучения студентов ИИТ РТУ МИРЭА программированию на языке Python [28, 29].

2.1. Тестовые наборы данных

В качестве тестовых наборов данных используются такие наборы, как синтетический трёхмерный набор s-curve.csv [25], трёхмерный набор mammoth.csv [26], позволяющий визуализировать изображение мамонта, и набор данных fmnist_test.csv [27], содержащий изображения размером 28 на 28 пикселей в оттенках серого цвета для 10 типов модных товаров.

На рисунке 1 представлены визуализации наборов данных s-curve.csv и mammoth.csv в трёхмерном пространстве. Каждый из этих наборов данных представляет собой единое целое, то есть для них, в первую очередь, важно сохранение глобальной структуры данных. При этом для набора данных mammoth.csv важно и сохранение локальной структуры данных (по возможности).

Для понимания того, как те или иные части трёхмерного объекта, описываемого строками набора данных, отображаются в трёхмерном пространстве, на рисунке 1 выполнена раскраска частей объектов в разные цвета.

Для набора данных s-curve.csv раскраска точек данных выполняется на основе автоматически сгенерированного массива вещественных чисел, число элементов в котором совпадает с числом наблюдений в синтетическом трёхмерном наборе. Для набора данных mammoth.csv число кластеров выбрано равным 10, а разбиение на кластеры получено с применением алгоритма агломеративной кластеризации. При этом раскраска точек данных выполняется в соответствии с их принадлежностью тому или иному кластеру.

а б

Рисунок 1. Визуализация наборов данных в трёхмерном пространстве:
а – s-curve.csv;
б – mammoth.csv

Для набора данных fmnist_test.csv характерна кластерная структура, позволяющая сгруппировать изображения в кластеры объектов одного типа. В связи с этим раскраска точек данных выполняется в соответствии с их принадлежностью тому или иному кластеру.

В дальнейшем информация по раскраске точек данных в исходном многомерном пространстве может быть использована при визуализации наборов данных, вложенных в двухмерное пространство алгоритмом PaCMAP.

В таблице 1 представлена информация по используемым тестовым наборам данных (если число классов для набора данных не задано, в соответствующей ячейке стоит знак «–»). При выполнении экспериментальных исследований с набором данных mammoth.csv рассматриваются только 10000 случайно выбранных строк (ввиду большого размера исходного набора данных, содержащего 999778 строк данных).

Таблица 1. Информация по тестовым наборам данных

Название набора данных	Число признаков (размерность набора данных)	Число классов	Число наблюдений (число строк в наборе данных)
s-curve.csv	3	–	1500
mammoth.csv	3	–	999778 (в экспериментах используется выборка, содержащая 10000 случайно выбранных строк исходного набора данных)
fmnist_test.csv	784 (28·28)	10	10000

2.2. Практические наборы данных

В качестве практических наборов данных используются три набора данных, сформированных при решении задач в сфере онлайн-обучения студентов 2 курса ИИТ РТУ МИРЭА программированию на языке Python [28, 29]. Эти наборы данных содержат уже обработанную информацию, извлеченную из лог-журналов событий системы «Цифрового ассистента преподавателя» (ЦАП) по завершении весенних семестров в 2023, 2024 и 2025 годах.

Лог-журналы событий в ЦАП в течении весенних семестров в 2023, 2024 и 2025 годах накапливали информацию о результатах решения студентами 2 курса ИИТ РТУ МИРЭА индивидуальных упражнений разной сложности по ряду задач, охватывающих различные разделы программирования на языке Python. При этом в 2023 и 2025 годах студентам были предложены индивидуальные упражнения по 11 задачам, а в 2024 году – по 12 задачам.

Предварительно из лог-журналов событий по завершении весенних семестров в 2023, 2024 и 2025 годах была извлечена информация, представленная соответственно в виде csv-файлов (наборов данных) messages_2023.csv, messages_2024.csv и messages_2025.csv со столбцами, озаглавленными как:

- group (группа студента);
- task (задача);

- variant (вариант);
- time (время отправки решения упражнения по задаче);
- status (состояние решения упражнения по задаче);
- student (ID студента);
- achievement (номер способа решения упражнения по задаче).

Таким образом, каждая строка в каждом csv-файле содержала информацию о результатах контакта конкретного студента с системой ЦАП при решении индивидуального упражнения по той или иной задаче.

При этом в столбце status могли находиться такие значения, как 2 (если решение принято), 3 (если решение отклонено), 6 (если было отклонено новое решение по задаче, уже решённой ранее). Если значение в столбце status было равно 2, то в столбце achievement отображался номер способа решения индивидуального упражнения по соответствующей задаче. Следует отметить, что заинтересованные студенты могли пытаться найти все известные системе ЦАП способы решения индивидуального упражнения по соответствующей задаче. Информация о таких достижениях студентов сохранялась в лог-журнале событий системы ЦАП.

Условием допуска студента к промежуточной аттестации в форме зачета была успешная сдача решений упражнений по всем заявленным задачам. Поэтому не все студенты, начавшие обучение, были допущены к промежуточной аттестации в форме зачета.

Существенный интерес представляет анализ данных о студентах, допущенных промежуточной аттестации в форме зачета. В частности, было бы интересно выявить в этих данных кластеры, то есть группы студентов со схожим поведением в процессе обучения [29, 30].

Полезным инструментом предварительного анализа может оказаться алгоритм нелинейного снижения размерности, например, алгоритм PaCMAP. В связи с этим в результате обработки информации из наборов данных messages_2023.csv, messages_2024.csv и messages_2025.csv были сформированы наборы данных results_2023.csv, results_2024.csv и results_2025.csv соответственно, содержащие столбцы, характеризующие:

- время начала решения индивидуального упражнения по каждой задаче (число столбцов равно числу задач);
- число неудачных попыток до первой удачной при решении индивидуального упражнения по каждой задаче (число столбцов равно числу задач);
- длительность решения (то есть время от первой попытки до первой удачной) индивидуального упражнения по каждой задаче (число столбцов равно числу задач);
- общее число достижений по всем задачам.

Таким образом, каждая строка в каждом новом csv-файле содержала информацию о результатах обучения конкретного студента, допущенного к промежуточной аттестации в форме зачета, в течение весеннего семестра.

Так как система ЦАП предлагала к решению 11 задач в 2023 и 2025 годах, а в 2024 год – 12 задач, то число столбцов, определяющих признаки в наборах данных на основе файлов results_2023.csv, results_2025.csv и results_2025.csv, оказалось равным 34 (11+11+11+1), 37 (12+12+12+1) и 34 (11+11+11+1) соответственно.

В таблице 2 представлена информация по используемым наборам данных при решении практических задач в сфере онлайн-обучения студентов. Так как число классов для этих наборов данных не известно, в соответствующих ячейках стоит знак «–». Число студентов, допущенных к промежуточной аттестации в форме зачета в 2023, 2024 и 2025 годах, отражено в последнем столбце таблицы 2.

Таблица 2. Информация по практическим наборам данных при решении задач в сфере онлайн-обучения студентов

Название набора данных	Число признаков (размерность набора данных)	Число классов	Число наблюдений (число строк в наборе данных)
results_2023.csv	34	–	1056
results_2024.csv	37	–	1111
results_2025.csv	34	–	1080

Для понимания того, как близкие (в исходном многомерном пространстве) точки данных, описываемые строками каждого набора данных, отображаются в двухмерном пространстве, число кластеров выбрано равным 5, а разбиение на кластеры получено с применением алгоритма агломеративной кластеризации.

В дальнейшем информация по раскраске точек данных в исходном многомерном пространстве может быть использована при визуализации наборов данных, вложенных в двухмерное пространство алгоритмом PaCMAP.

3. Предварительная обработка данных и инициализация вложений в пространство низкой размерности

В настоящей работе предлагается изучить целесообразность использования 11 алгоритмов снижения размерности в составе алгоритма PaCMAP при реализации различных сценариев предварительной обработки данных и инициализации вложений данных в двухмерное пространство.

В таблице 3 приведены названия используемых алгоритмов снижения размерности и значения их изменяемых параметров. Значения неуказанных в таблице 3 параметров заданы по умолчанию.

Таблица 3. Алгоритмы снижения размерности и значения их изменяемых параметров для экспериментальных наборов данных

№	Название алгоритма	Набор данных: изменяемые параметры и их значения
1	PCA	–
2	MDS	–
3	Isomap	• s_curve.csv: n_neighbors [5 (по умолчанию) и 10], где n_neighbors – число ближайших соседей • mammoth.csv: n_neighbors [5 (по умолчанию) и 10] • fmnist_test.csv: n_neighbors [5 (по умолчанию), 10 и 15] • results_2023.csv: n_neighbors [5 (по умолчанию) и 10] • results_2024.csv: n_neighbors [5 (по умолчанию) и 10] • results_2025.csv: n_neighbors [5 (по умолчанию) и 10]
4	SE	• s_curve.csv: n_neighbors [max(int(n/10), 1) (по умолчанию) и 10], где n – число наблюдений • mammoth.csv: n_neighbors [max(int(n/10), 1) (по умолчанию) и 10] • fmnist_test.csv: n_neighbors [max(int(n/10), 1) (по умолчанию) и 10] • results_2023.csv: n_neighbors [max(int(n/10), 1) (по умолчанию) и 10] • results_2024.csv: n_neighbors [max(int(n/10), 1) (по умолчанию) и 10] • results_2025.csv: n_neighbors [max(int(n/10), 1) (по умолчанию) и 10]
5	SLLE	• s_curve.csv: n_neighbors [5 (по умолчанию) и 10] • mammoth.csv: n_neighbors [5 (по умолчанию) и 10] • fmnist_test.csv: n_neighbors [5 (по умолчанию), 10 и 150] • results_2023.csv: n_neighbors [5 (по умолчанию) и 10] • results_2024.csv: n_neighbors [5 (по умолчанию) и 10] • results_2025.csv: n_neighbors [5 (по умолчанию) и 10]
6	MLLE	• s_curve.csv: n_neighbors [5 (по умолчанию) и 10] • mammoth.csv: n_neighbors [5 (по умолчанию) и 10] • fmnist_test.csv: n_neighbors [5 (по умолчанию), 10 и 150] • results_2023.csv: n_neighbors [5 (по умолчанию) и 10] • results_2024.csv: n_neighbors [5 (по умолчанию) и 10] • results_2025.csv: n_neighbors [5 (по умолчанию) и 10]
7	LLE с алгоритмом HE	• s_curve.csv: n_neighbors [6 и 10], eigen_solver='dense', где eigen_solver определяет метод, используемый для вычисления собственных векторов (значение 'dense' определяет метод, предполагающий использование стандартных операций с плотными матрицами для разложения по собственным значениям). • mammoth.csv: n_neighbors [6 и 10], eigen_solver='dense' • fmnist_test.csv: n_neighbors [6, 10 и 150], eigen_solver='dense' • results_2023.csv: n_neighbors [6 и 10], eigen_solver='dense' • results_2024.csv: n_neighbors [6 и 10], eigen_solver='dense' • results_2025.csv: n_neighbors [6 и 10], eigen_solver='dense' Примечание: n_neighbors > n_components · (n_components + 3) / 2 [7], где n_components – число измерений в пространстве низкой размерности. Так как n_components=2, то n_neighbors > 5.

Окончание таблицы 3

№	Название алгоритма	Набор данных: изменяемые параметры и их значения
8	LLE с алгоритмом LTSA	• s_curve.csv: n_neighbors [5 (по умолчанию) и 10] • mammoth.csv: n_neighbors [5 (по умолчанию) и 10] • fmnist_test.csv: n_neighbors [5 (по умолчанию), 10 и 150] • results_2023.csv: n_neighbors [5 (по умолчанию) и 10] • results_2024.csv: n_neighbors [5 (по умолчанию) и 10] • results_2025.csv: n_neighbors [5 (по умолчанию) и 10]
9	t-SNE	• perplexity (that is related to the number of nearest neighbors): • s_curve.csv: perplexity [15 и 30 (по умолчанию)] где perplexity – параметр, сопоставимый с числом ближайших соседей n_neighbors • mammoth.csv: perplexity [15 и 30 (по умолчанию)] • fmnist_test.csv: perplexity [10, 15 и 30 (по умолчанию)] • results_2023.csv: perplexity [15 и 30 (по умолчанию)] • results_2024.csv: perplexity [15 и 30 (по умолчанию)] • results_2025.csv: perplexity [15 и 30 (по умолчанию)]
10	UMAP	• s_curve.csv: n_neighbors [10 и 15 (по умолчанию)] • mammoth.csv: n_neighbors [10 и 15 (по умолчанию)] • fmnist_test.csv: n_neighbors [10, 15 (по умолчанию) и 30] • results_2023.csv: n_neighbors [10 и 15 (по умолчанию)] • results_2024.csv: n_neighbors [10 и 15 (по умолчанию)] • results_2025.csv: n_neighbors [10 и 15 (по умолчанию)]
11	TriMAP	• s_curve.csv: n_inliers [10 и 12 (по умолчанию)], n_inliers – число ближайших соседей для формирования триплетов ближайших соседей • mammoth.csv: n_inliers [10 и 12 (по умолчанию)] • fmnist_test.csv: n_inliers [10 и 12 (по умолчанию)] • results_2023.csv: n_inliers [10 и 12 (по умолчанию)] • results_2024.csv: n_inliers [10 и 12 (по умолчанию)] • results_2025.csv: n_inliers [10 и 12 (по умолчанию)]

Как видно из таблицы 3, значения изменяемых параметров для наборов данных s_curve.csv, mammoth.csv, results_2023.csv, results_2024.csv и results_2025.csv предлагается выбирать одинаковым образом. Для набора данных fmnist_test.csv в ряде случаев добавлены дополнительные значения параметров в виду большей размерности анализируемого набора данных и, в частности, специфики работы с алгоритмом LLE. При этом, устанавливая в различных алгоритмах снижения размерности (таблица 3) число ближайших соседей n_neighbors равным 10 (n_neighbors=10), число ближайших соседей n_inliers для формирования триплетов ближайших соседей равным 10 (n_inliers=10), а также изменяя значение параметра perplexity, можно попытаться сопоставить значения этих параметров значению параметра n_neighbors в алгоритме PaCMAP, равному по умолчанию 10.

Следует отметить, что различные алгоритмы снижения размерности используют разные функции потерь, ориентированные на достижение целей, поставленных перед алгоритмами. В связи с этим сравнение результатов работы различных алгоритмов на основе значений их функций потерь не представляется осуществимым – возможно только визуальное сравнение результатов работы алгоритмов снижения размерности. При этом даже в рамках одного алгоритма использование различных функций потерь приводит к разным результатам [18, 22, 23].

Для каждого набора данных предлагается рассмотреть 4 сценария работы алгоритма PaCMAP:

- случайная инициализация (Random Initialization, RI) вложения в двухмерное пространство и отказ от предварительной обработки (Refusal of Preprocessing, RoP) набора данных (RI/RoP);
- алгоритмическая инициализация (Algorithmic Initialization, AI) вложения в двухмерное пространство на основе некоторого алгоритма снижения размерности (таблица 3) и отказ от предобработки набора данных (AI/RoP);
- случайная инициализация вложения в двухмерное пространство и алгоритмическая предобработка (Algorithmic Preprocessing, AP) набора данных на основе некоторого алгоритма снижения размерности (таблица 3) (RI/AP);

- алгоритмическая инициализация вложения в двухмерное пространство на основе некоторого алгоритма снижения размерности (таблица 3) и алгоритмическая предобработка набора данных на основе некоторого алгоритма снижения размерности (таблица 3) (AI/AP).

В настоящей работе при совместном использовании как предварительной обработки данных, так и алгоритмической инициализации вложения данных в двухмерное пространство на основе некоторого алгоритма снижения размерности предлагается использовать один и тот же алгоритм снижения размерности из таблицы 1.

При этом качество работы алгоритма PaCMAP предполагается оценивать визуально и с точки зрения минимизации функции потерь. Кроме того, предлагается оценивать время работы алгоритма PaCMAP: время должно быть минимизировано.

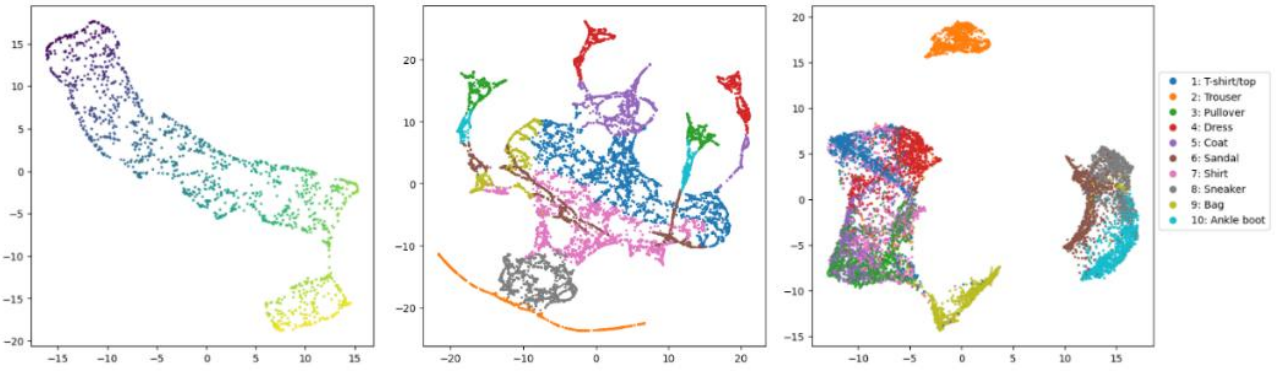
4. Экспериментальные исследования

Экспериментальные исследования на тестовых и практических наборах данных выполнены на языке Python 3.10 в среде Google Colab с применением программной реализации алгоритма PaCMAP [24] и библиотек машинного обучения, реализующих различные дополнительно используемые алгоритмы снижения размерности наряду с алгоритмом PCA, предлагаемым к использованию в алгоритме PaCMAP.

4.1. Экспериментальные исследования на тестовых наборах данных

Экспериментальные исследования на тестовых наборах данных выполнены с применением четырех сценариев (п. 3) работы алгоритма PaCMAP [31].

На рисунке 2 представлена визуализация результатов работы алгоритма PaCMAP для трёх тестовых наборов данных, когда осуществлены случайная инициализация вложения набора данных в двухмерное пространство с использованием нормального распределения и отказ от предобработки набора данных (по сценарию RI/RoP). При этом для каждого тестового набора данных указаны значения функции потерь и время расчетов в формате Loss/время. Из рисунка 2,б видно, что вложение трёхмерного изображения мамонта (рисунок 1, б) в двухмерное пространство представляет собой проекцию, которую можно условно назвать «вид снизу». Визуально можно хорошо различить голову мамонта, бивни, задние конечности. При этом изображение (в целом), в том числе – расположение передних конечностей, оставляет желать лучшего. К изображению в двухмерном пространстве (рисунок 2,а) трёхмерного объекта, представленного на рисунке 1,а есть большие вопросы. На рисунке 2,в можно заметить удачно выделенный кластер «Trouser» и неплохую группировку похожих между собой модных товаров.



Loss/время=2514.278320/0.65 с Loss/время=12826.658203/3.72 с Loss/время=18363.742188/4.43 с

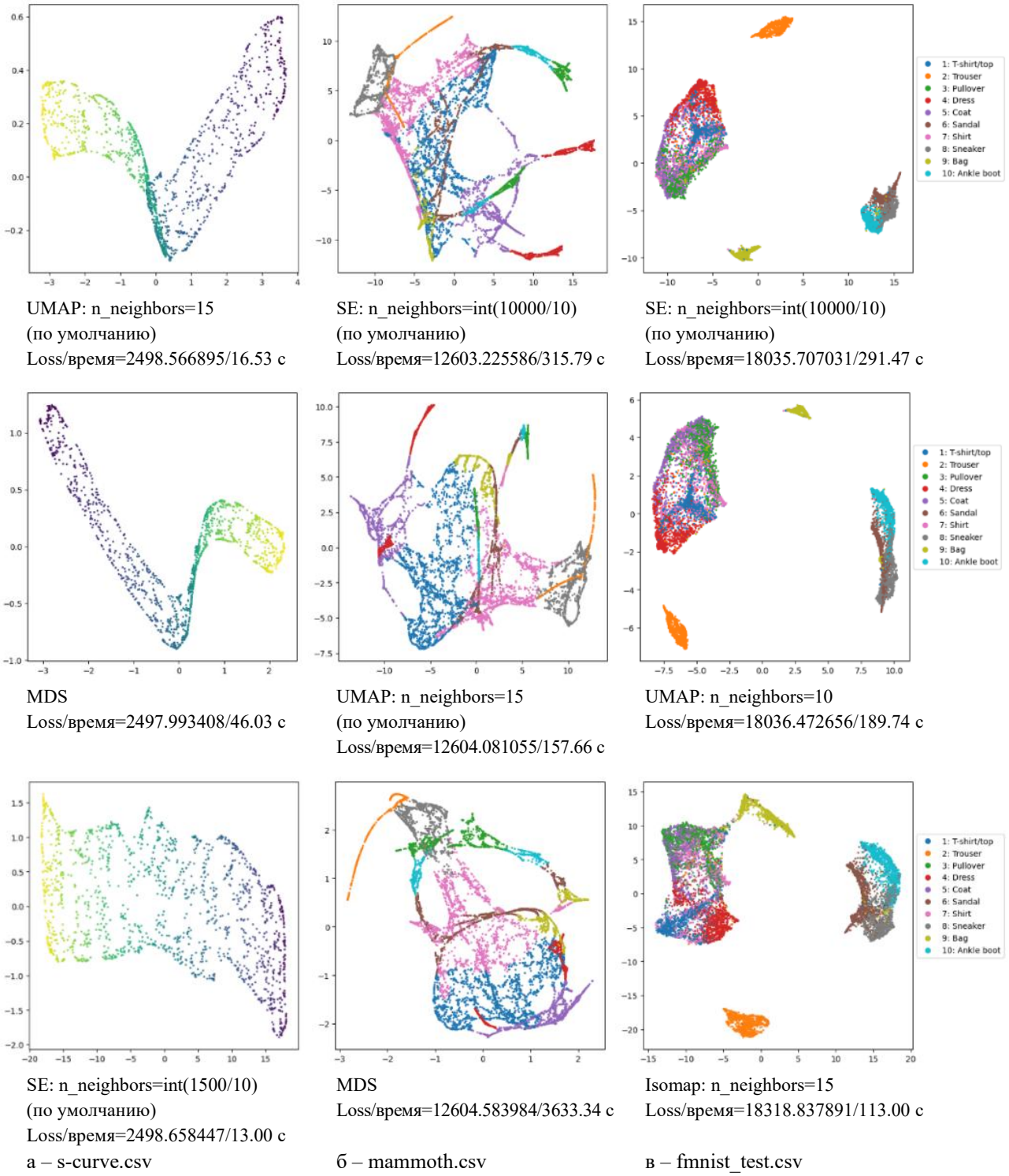
а – s-curve.csv б – mammoth.csv в – fmnist_test.csv

Рисунок 2. Визуализация для тестовых наборов данных результатов работы алгоритма PaCMAP, когда осуществлены случайная инициализация вложения в двухмерное пространство с использованием нормального распределения и отказ от предобработки набора данных (по сценарию RI/RoP)

По результатам анализа значений функции потерь и визуального анализа представленных изображений возникает вопрос: можно ли улучшить результаты вложения рассматриваемых наборов данных в двумерное пространство?

На рисунке 3 для каждого тестового набора данных представлена визуализация 3 лучших результатов работы алгоритма PaCMAP (в смысле минимизации функции потерь), когда осуществлены алгоритмическая инициализация вложения набора данных в двухмерное пространство с использованием одного из 11 алгоритмов снижения размерности (таблица 3) и отказ от предобработки набора данных (по сценарию AI/RoP). При этом для каждого результата работы алгоритма PaCMAP указаны значения функции потерь и время

расчетов в формате Loss/время, а также соответствующие им значения изменяемых параметров (таблица 3). Из рисунка 3 видно, что применение алгоритмов снижения размерности для алгоритмической инициализации вложений всех трёх тестовых наборов данных позволило улучшить результаты работы алгоритма PaCMAP (в смысле минимизации функции потерь). При этом алгоритм PCA для рассматриваемых наборов данных не вошел в тройку лидеров. Следует отметить, однако, что визуально 3 лучших результата работы алгоритма PaCMAP на наборе данных mammoth.csv (рисунок 3,б) выглядят менее качественно, чем на рисунке 2,б. Возможно, этот факт можно объяснить как сложностью организации самого набора данных и малым объёмом используемой выборки (низкой плотностью точек данных), так и целесообразностью дальнейшей доработки самой функции потерь.



UMAP: n_neighbors=15 (по умолчанию) Loss/время=2498.566895/16.53 с

SE: n_neighbors=int(10000/10) (по умолчанию) Loss/время=12603.225586/315.79 с

SE: n_neighbors=int(10000/10) (по умолчанию) Loss/время=18035.707031/291.47 с

MDS Loss/время=2497.993408/46.03 с

UMAP: n_neighbors=15 (по умолчанию) Loss/время=12604.081055/157.66 с

UMAP: n_neighbors=10 Loss/время=18036.472656/189.74 с

SE: n_neighbors=int(1500/10) (по умолчанию) Loss/время=2498.658447/13.00 с

MDS Loss/время=12604.583984/3633.34 с

Isomap: n_neighbors=15 Loss/время=18318.837891/113.00 с

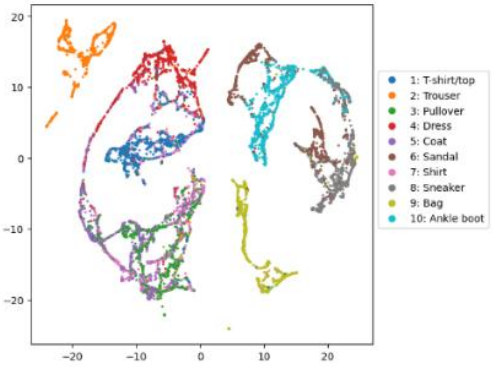
а – s-curve.csv

б – mammoth.csv

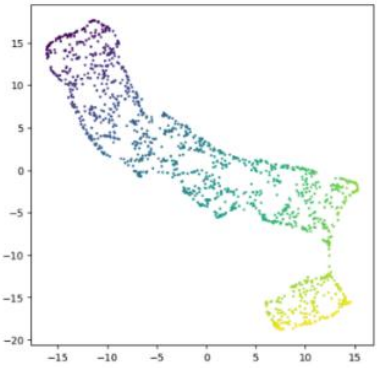
в – fmnist_test.csv

Рисунок 3. Визуализация для тестовых наборов данных лучших результатов работы алгоритма PaCMAP, когда осуществлены алгоритмическая инициализация вложения в двухмерное пространство с использованием одного из 11 алгоритмов снижения размерности и отказ от предобработки набора данных (по сценарию AI/RoP)

На рисунке 4 для каждого тестового набора данных представлена визуализация лучших результатов работы алгоритма PaCMAP (в смысле минимизации функции потерь), когда осуществлены случайная инициализация вложения в двухмерное пространство с использованием нормального распределения и алгоритмическая предобработка набора данных с использованием одного из 11 алгоритмов снижения размерности (по сценарию RI/AP).

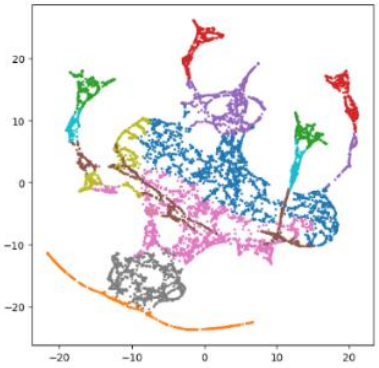


UMAP: n_neighbors=10
Loss/время=12723.467773/5.33 с



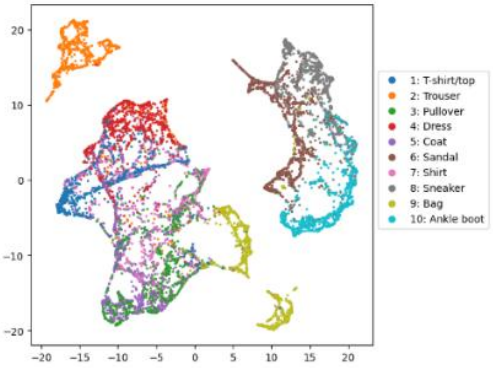
PCA
Loss/время=2514.278320/0.82 с

(результаты для функции потерь одинаковые для всех алгоритмов, небольшие различия имеются только во времени реализации)

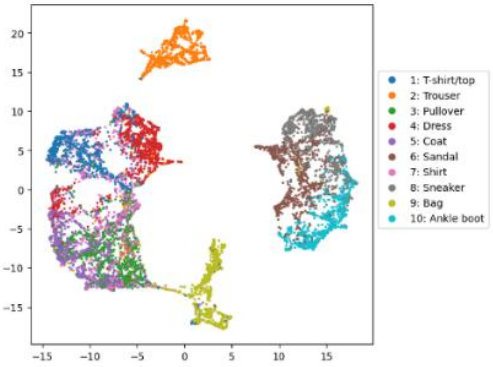


PCA
Loss/время=12826.658203/5.45 с

(результаты для функции потерь одинаковые для всех алгоритмов, небольшие различия имеются только во времени реализации)



TriMAP: n_inliers=12
Loss/время=13368.562500/3.69 с



Isomap: n_neighbors=5
(по умолчанию)
Loss/время=16278.180664/5.09 с

а – s-curve.csv

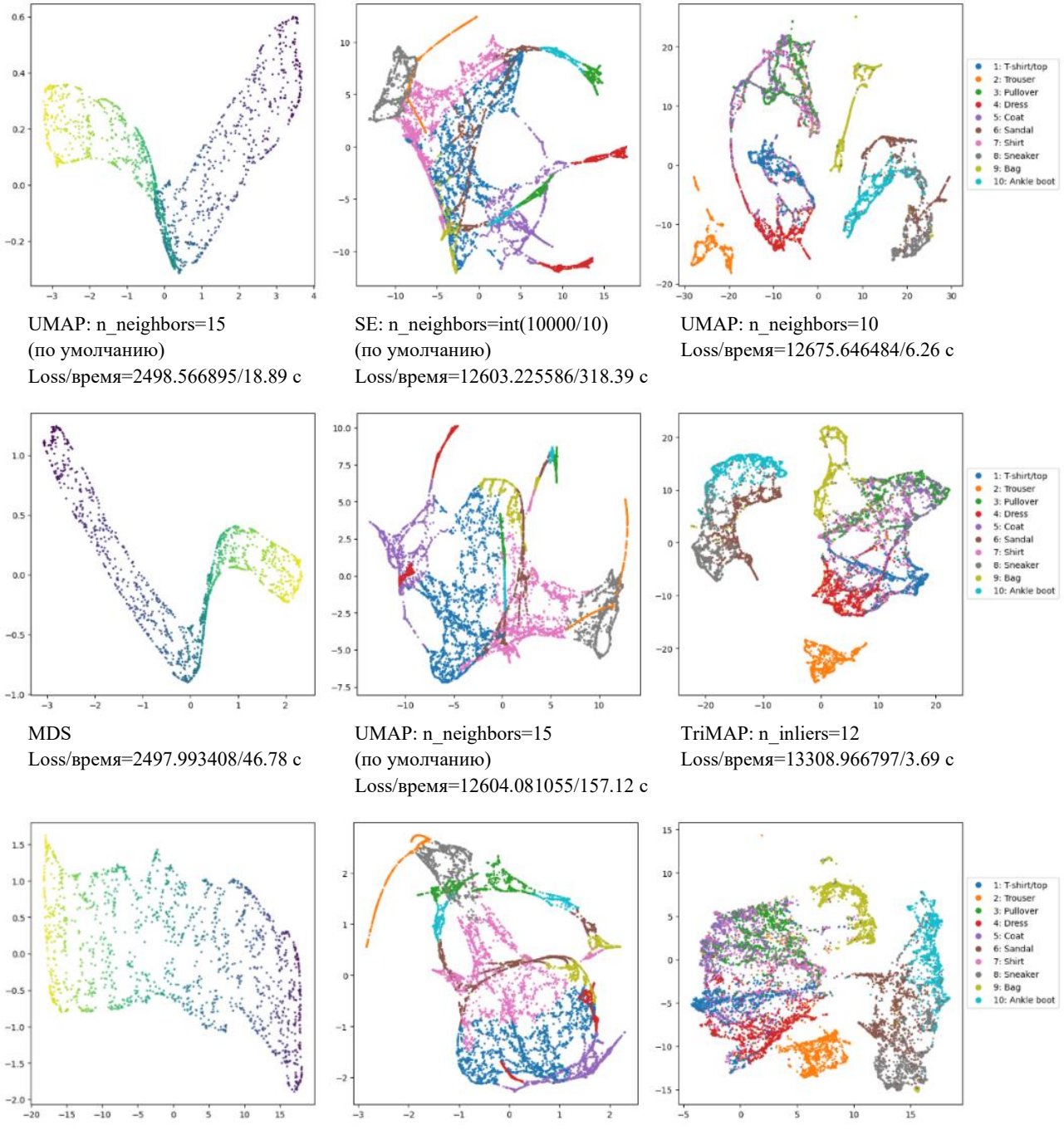
б – mammoth.csv

в – fmnist_test.csv

Рисунок 4. Визуализация для тестовых наборов данных лучших результатов работы алгоритма PaCMAP, когда осуществлены случайная инициализация вложения в двухмерное пространство с использованием нормального распределения и алгоритмическая предобработка набора данных с использованием одного из 11 алгоритмов снижения размерности (по сценарию RI/AP)

Так как для наборов данных s-curve.csv и mammoth.csv оказалось, что результаты работы алгоритма PaCMAP не зависят от выбора алгоритма снижения размерности для предобработки набора данных (результаты для функции

потерь одинаковые для всех алгоритмов снижения размерности), то на рисунках 4,а и 4,б представлены результаты, полученные при применении алгоритма PCA для предобработки рассматриваемого набора данных. При этом небольшие различия были выявлены только во времени реализации алгоритмов снижения размерности. Совпадение значений для функций потерь при использовании разных алгоритмов снижения размерности для предобработки этих наборов данных можно объяснить их малой размерностью: каждый набор содержит три признака (таблица 1).



UMAP: n_neighbors=15
(по умолчанию)
Loss/время=2498.566895/18.89 с

SE: n_neighbors=int(10000/10)
(по умолчанию)
Loss/время=12603.225586/318.39 с

UMAP: n_neighbors=10
Loss/время=12675.646484/6.26 с

MDS
Loss/время=2497.993408/46.78 с

UMAP: n_neighbors=15
(по умолчанию)
Loss/время=12604.081055/157.12 с

TriMAP: n_inliers=12
Loss/время=13308.966797/3.69 с

SE: n_neighbors =int(1500/10)
(по умолчанию)
Loss/время=2498.658447/13.30 с

MDS
Loss/время=12604.583984/3637.94 с

Isomap: n_neighbors =5
(по умолчанию)
Loss/время=17217.468750/4.58 с

а – s-curve.csv

б – mammoth.csv

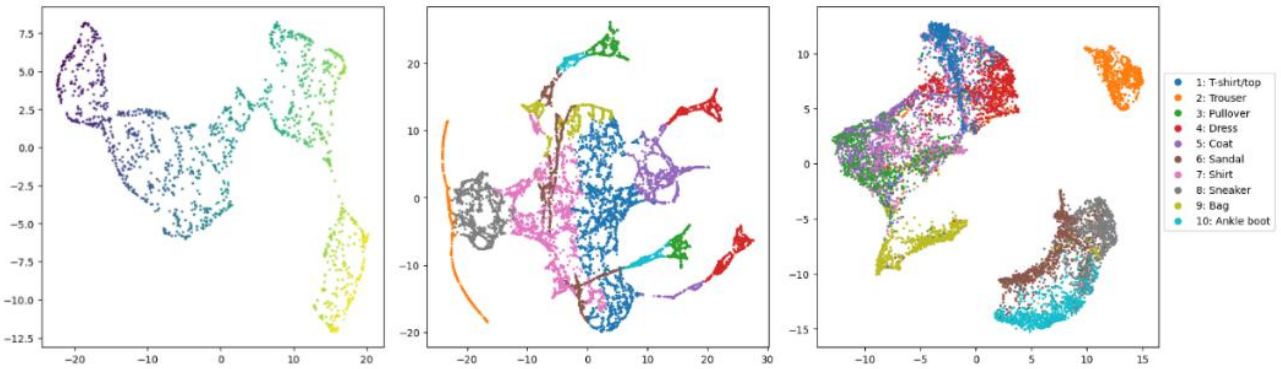
в – fmnist_test.csv

Рисунок 5. Визуализация для тестовых наборов данных лучших результатов работы алгоритма PaCMAP, когда осуществлены алгоритмическая инициализация вложения в двухмерное пространство и алгоритмическая предобработка набора данных с использованием одного из 11 алгоритмов снижения размерности (по сценарию AI/AP)

Для набора данных fmnist_test.csv на рисунке 4,в представлена визуализация 3 лучших результатов работы алгоритма PaCMAP (в смысле минимизации функции потерь). Этот набор данных многомерный: каждое наблюдение описывается 784 признаками (таблица 1). Для каждого результата работы алгоритма PaCMAP указаны значения функции потерь и время расчетов в формате Loss/время, а также соответствующие им значения изменяемых параметров (таблица 3). Из рисунка 4,в видно, что применение алгоритмов снижения размерности для предобработки набора данных fmnist_test.csv позволило улучшить результаты работы алгоритма PaCMAP (в смысле минимизации функции потерь). При этом алгоритм PCA для этого набора данных не вошел в тройку лидеров.

На рисунке 5 для каждого тестового набора данных представлена визуализация 3 лучших результатов работы алгоритма PaCMAP (в смысле минимизации функции потерь), когда осуществлены алгоритмическая инициализация вложения в двухмерное пространство и алгоритмическая предобработка набора данных с использованием одного из 11 алгоритмов снижения размерности (по сценарию AI/AP). Для каждого результата работы алгоритма PaCMAP указаны значения функции потерь и время расчетов в формате Loss/время, а также соответствующие им значения изменяемых параметров (таблица 3). Для наборов данных s-curve.csv и mammoth.csv результаты работы алгоритма PaCMAP по сценарию AI/AP (рисунки 5,а и 5,б соответственно) полностью совпали с результатами работы по сценарию по сценарию AI/RoP (рисунки 3,а и 3,б соответственно) в смысле полученных значений функции потерь при применении различных алгоритмов снижения размерности. При этом небольшие различия были выявлены только во времени реализации алгоритмов снижения размерности. Совпадение значений для функций потерь при использовании и отказе от использования алгоритмов снижения размерности для предобработки этих наборов данных можно объяснить их малой размерностью: каждый набор содержит три признака (таблица 1). Для набора данных fmnist_test.csv следует отметить существенные изменения (рисунок 5,в) как в перечне названий алгоритмов снижения размерности для 3 лучших результатов работы алгоритма PaCMAP (в смысле минимизации функции потерь), так и в самих значениях функций потерь: они стали значительно меньше. Так как набор данных fmnist_test.csv является многомерным, использование алгоритмов снижения размерности для предобработки этого набора данных оказало существенное влияние на результаты оптимизации.

На рисунке 6 для каждого тестового набора данных представлена визуализация результатов работы алгоритма PaCMAP при использовании в его составе алгоритма PCA, как это предложено авторами [15], по сценарию AI/AP. Как уже было отмечено ранее, алгоритм PCA не вошел в число лидеров алгоритмов снижения размерности (таблица 3), применяемых в составе алгоритма PaCMAP по сценариям AI/AP, RI/AP и AI/RoP, в смысле минимизации функции потерь. При этом достоинство алгоритма PCA заключается в минимальных временных затратах на реализацию алгоритма PaCMAP при включении алгоритма PCA в его состав.



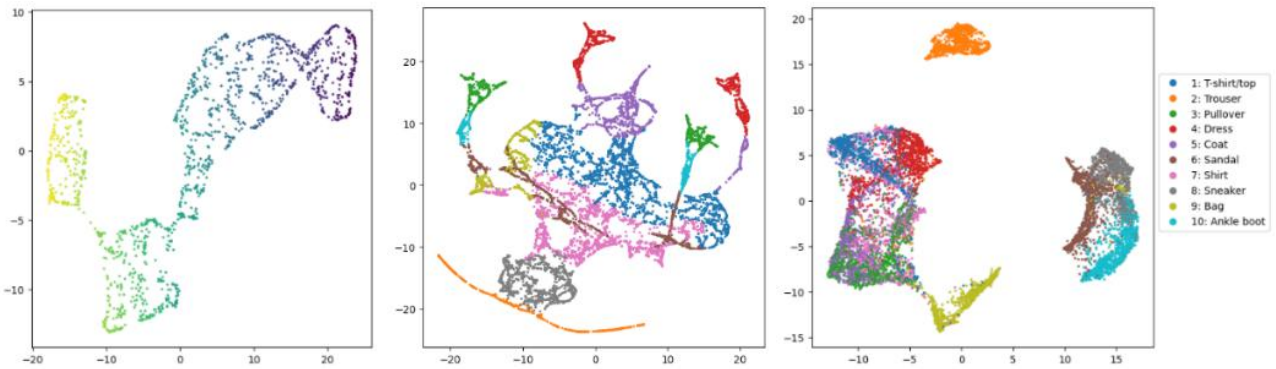
PCA/PCA Loss/время=2547.701416/0.82 с	PCA/PCA Loss/время=12808.285156/3.70 с	PCA/PCA Loss/время=18267.097656/4.45 с
а – s-curve.csv	б– mammoth.csv	в – fmnist_test.csv

Рисунок 6. Визуализация для тестовых наборов данных результатов работы алгоритма PaCMAP при использовании в его составе алгоритма PCA, как это предложено авторами в [15], по сценарию AI/AP

На рисунке 7 для каждого тестового набора данных представлена визуализация худших результатов работы алгоритма PaCMAP при использовании в его составе различных алгоритмов снижения размерности. Для каждого результата указаны значения функции потерь и время расчетов в формате Loss/время.

По результатам экспериментальных исследований на тестовых наборах данных можно сделать следующие выводы.

Алгоритм PaCMAP в авторской версии для набора данных s-curve.csv по сценарию AI/AP отработал хуже в смысле минимизации функции потерь (рисунок 6,а), чем при использовании других алгоритмов снижения размерности [за исключением алгоритма SLLE (рисунок 7,а)]. Аналогичные выводы справедливы и при сравнении результатов работы алгоритма PaCMAP для набора данных s-curve.csv по сценарию AI/RoP. При этом использование алгоритма PCA в составе алгоритма PaCMAP привело к худшим результатам работы, чем в случае применения сценариев RI/RoP (рисунок 2,а) и RI/AP (рисунок 4,а). Алгоритм PaCMAP в авторской версии для набора данных mammoth.csv по сценариям AI/RoP и AI/AP (рисунок 6,б) отработал лучше, чем по сценариям RI/RoP (рисунок 2,б) и RI/AP (рисунок 4,б). Алгоритм PaCMAP в авторской версии для набора данных fmnist_test.csv по сценариям AI/RoP и AI/AP (рисунок 6,в) отработал лучше, чем по сценариям RI/RoP (рисунок 2,в) и RI/AP (рисунок 4,в), но был превзойден некоторыми другими версиями алгоритма PaCMAP, использующими алгоритмы снижения размерности, отличные от алгоритма PCA (рисунки 3,в и 5,в).



SLLE: n_neighbors=5 Loss/время=2548.256348/3.08 с (по сценарию AI/AP)	Loss/время=12826.658203/3.72 с (время для PCA) (по сценарию RI/RoP)	Loss/время=18363.742188/4.43 с (время для PCA) (по сценарию RI/RoP)
SLLE: n_neighbors =5 Loss/время=2548.256348/3.29 с (по сценарию AI/RoP)	Loss/время=12826.658203/5.45 с (время для PCA) (по сценарию RI/AP)	
а – s-curve.csv	б – mammoth.csv	в – fmnist_test.csv

Рисунок 7. Визуализация для тестовых наборов данных худших результатов работы алгоритма PaCMAP

Следует отметить, что многие алгоритмы снижения размерности (таблица 3) испытывали затруднения при работе с набором данных fmnist_test.csv. Такие алгоритмы, как алгоритм MDS (со значениями параметров, заданными по умолчанию), все варианты алгоритма LLE (при выбранных значениях изменяемых параметров) и алгоритм t-SNE (при выбранных значениях изменяемых параметров) не справились с вложением набора данных fmnist_test.csv в двухмерное пространство при сценариях RI/AP и AI/AP). При этом алгоритмы LLE (с алгоритмом HE) и LLE (с алгоритмом LTSA) при выбранных значениях изменяемых параметров не справились и с вложением набора данных fmnist_test.csv в двухмерное пространство при сценарии AI/RoP. В целом, проблемы с вложением набора данных fmnist_test.csv в двухмерное пространство не возникли только при случайной инициализации вложения в двухмерное пространство и отказе от предобработки набора данных (то есть при работе по сценарию RI/RoP) при реализации алгоритма PaCMAP.

Многие алгоритмы снижения размерности используют в своей работе инициализацию случайными числами. Поэтому при разных запусках алгоритмов возможно получение несколько отличающихся результатов вложения в двухмерное пространство. Многократные запуски алгоритмов могут позволить получить лучшие результаты вложения. Но при ограничении на время решения задачи вложения набора данных в двухмерное пространство целесообразно выбрать лучший результат из уже полученных.

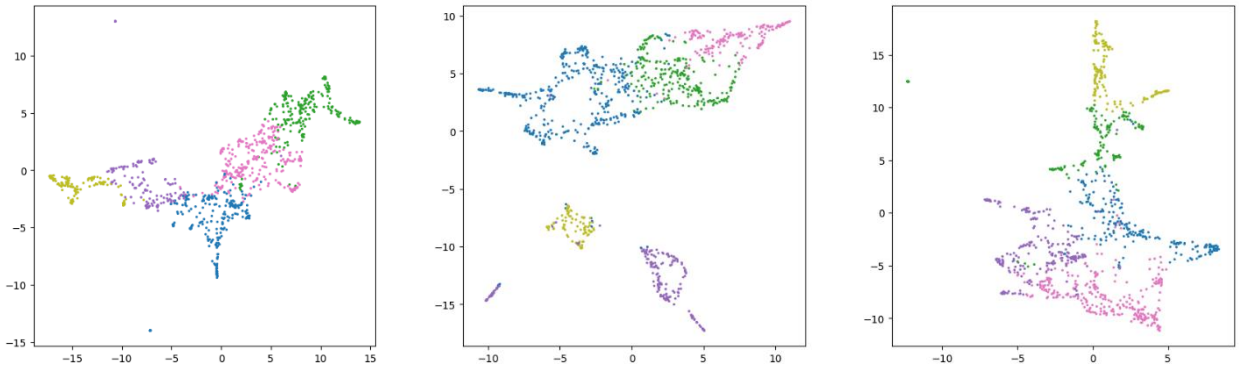
4.2. Экспериментальные исследования на практических наборах данных

Экспериментальные исследования на практических наборах данных выполнены с применением четырёх сценариев (п. 3) работы алгоритма PaCMAP.

При работе с тремя практическими наборами данных results_2023.csv, results_2024.csv и results_2025.csv, сформированными при решении задач в сфере онлайн-обучения студентов 2 курса ИИТ РТУ МИРЭА программированию на языке Python, были получены следующие результаты.

Для каждого практического набора данных использование сценариев RI/RoP и RI/AP привело к получению одинаковых результатов вложения (в смысле минимизации функции потерь), то есть результаты применения алгоритма PaCMAP, когда осуществлены случайная инициализация вложения в двухмерное пространство с использованием нормального распределения и отказ от предобработки набора данных, полностью совпадали с результаты применения алгоритма PaCMAP, когда осуществлены случайная инициализация вложения в двухмерное пространство с использованием нормального распределения и алгоритмическая предобработка набора данных с использованием одного из 11 алгоритмов снижения размерности, то есть использование алгоритмической предобработки набора данных не отразилось на результатах применения алгоритма PaCMAP.

На рисунке 8 для каждого практического набора данных представлена визуализация результатов работы алгоритма PaCMAP по сценарию RI/RoP.



RI/RoP
Loss/время=2545.664307/0.46 с
а – results_2023.csv

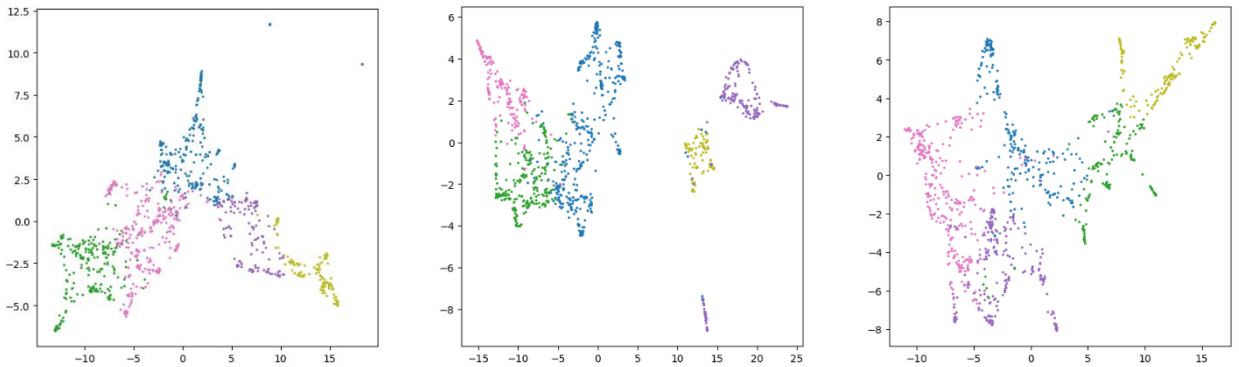
RI/RoP
Loss/время=2654.796143/0.44 с
б– results_2024.csv

RI/RoP
Loss/время=2614.044434/0.96 с
в – results_2025.csv

Рисунок 8. Визуализация для практических наборов данных результатов работы алгоритма PaCMAP по сценарию RI/RoP

Для каждого практического набора данных использование сценариев AI/RoP и AI/AP позволило получить результаты применения алгоритма PaCMAP, отличные от результатов применения алгоритма PaCMAP по сценариям RI/RoP и RI/AP. При этом существенную роль оказало только применение алгоритмической инициализации (AI) вложения в двухмерное пространство с использованием одного из 11 алгоритмов снижения размерности. Выбор способа предобработки набора данных (RoP или AP) не отразился на результатах применения алгоритма PaCMAP.

На рисунке 9 для каждого практического набора данных представлена визуализация результатов работы алгоритма PaCMAP при использовании в его составе алгоритма PCA, как это предложено авторами [15], по сценарию AI/RoP.



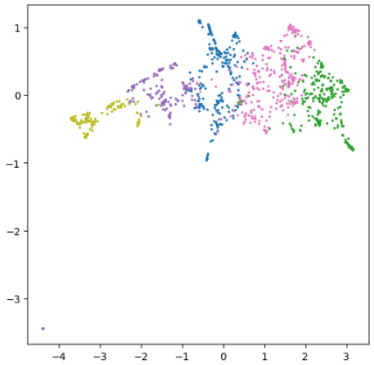
PCA/RoP
Loss/время=2543.050781/0.45 с
а – results_2023.csv

PCA/RoP
Loss/время=2664.572754 /0.47 с
б– results_2024.csv

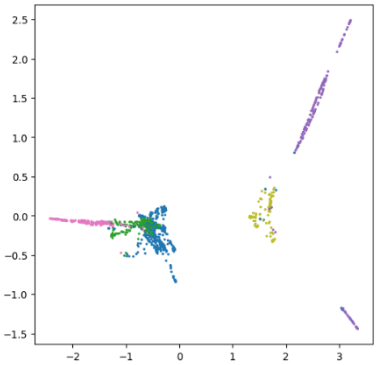
PCA/RoP
Loss/время=2612.301758 /0.76 с
в – results_2025.csv

Рисунок 9. Визуализация для практических наборов данных результатов работы алгоритма PaCMAP при использовании в его составе алгоритма PCA, как это предложено авторами [15], по сценарию AI/RoP

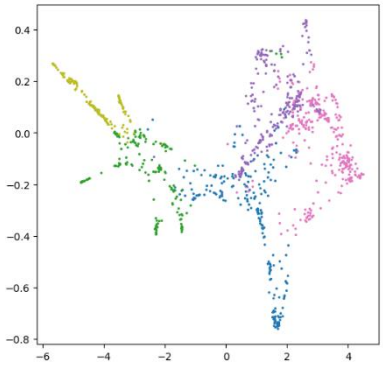
На рисунке 10 для каждого практического набора данных представлена визуализация 3 лучших результатов работы алгоритма PaCMAP (в смысле минимизации функции потерь), когда осуществлены алгоритмическая инициализация вложения набора данных в двухмерное пространство с использованием одного из 11 алгоритмов снижения размерности (таблица 3) и отказ от предобработки набора данных (по сценарию AI/RoP). При этом для каждого результата работы алгоритма PaCMAP указаны значения функции потерь и время расчетов в формате Loss/время, а также соответствующие им значения изменяемых параметров (таблица 3). Из рисунка 10 видно, что применение алгоритмов снижения размерности для алгоритмической инициализации вложений всех трёх тестовых наборов данных позволило улучшить результаты работы алгоритма PaCMAP (в смысле минимизации функции потерь). При этом алгоритм PCA для рассматриваемых наборов данных не вошел в тройку лидеров.



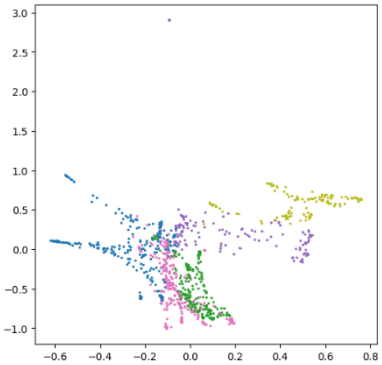
UMAP: n_neighbors=15
(по умолчанию)
Loss/время=2504.296387 /9.35 с



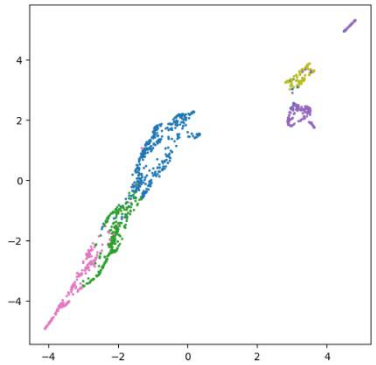
MDS
Loss/время= 2628.025391/70.64 с



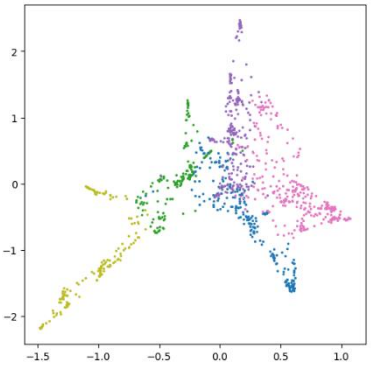
UMAP: n_neighbors=15
(по умолчанию)
Loss/время=2589.536133 /11.19 с



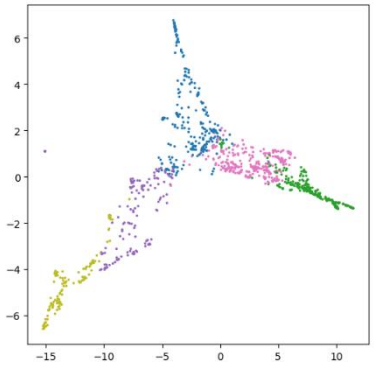
MDS
Loss/время=2504.279785 /48.23 с



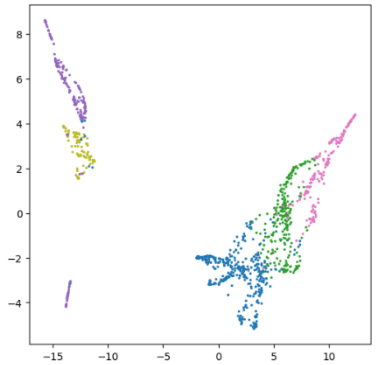
UMAP: n_neighbors=15
(по умолчанию)
Loss/время=2628.087646 /11.95 с



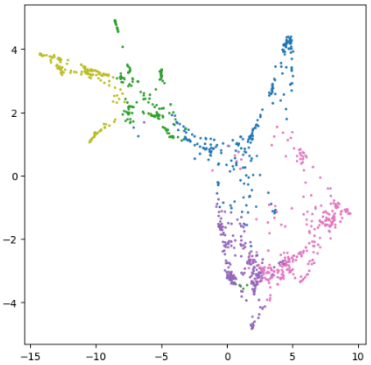
MDS
Loss/время=2589.538818/53.54 с



SE: n_neighbors=int(1056/10)
(по умолчанию)
Loss/время=2504.361816/7.96 с



SE: n_neighbors=int(1111/10)
(по умолчанию)
Loss/время=2628.457520/12.87 с



SE: n_neighbors=int(1080/10)
(по умолчанию)
Loss/время=2589.543945/7.34 с

а – results_2023.csv б – results_2024.csv в – results_2025.csv

Рисунок 10. Визуализация для практических наборов данных лучших результатов работы алгоритма PaCMAP, когда осуществлены алгоритмическая инициализация вложения в двухмерное пространство с использованием одного из 11 алгоритмов снижения размерности и отказ от предобработки набора данных (по сценарию AI/RoP)

Для набора данных results_2023.csv лучшие результаты (в смысле минимизации функции потерь) по сценарию AI/RoP показало применение алгоритмов UMAP, MDS и SE (перечисленных в порядке убывания значений функции потерь) в составе алгоритма PaCMAP.

Для набора данных results_2024.csv лучшие результаты (в смысле минимизации функции потерь) по сценарию AI/RoP показало применение алгоритмов MDS, UMAP и SE (перечисленных в порядке убывания значений функции потерь) в составе алгоритма PaCMAP.

Для набора данных results_2025.csv лучшие результаты (в смысле минимизации функции потерь) по сценарию AI/RoP показало применение алгоритмов UMAP, MDS и SE (перечисленных в порядке убывания значений функции потерь) в составе алгоритма PaCMAP.

Визуальный анализ рисунков 8 – 10 позволяет сделать следующие выводы. Вложение с использованием алгоритма PaCMAP в двухмерное пространство многомерных практических наборов данных, характеризующих активность студентов, допущенных к промежуточной аттестации в форме зачета в 2023, 2024 и 2025 годах, является успешным. Об этом можно судить по хорошему группированию в двухмерном пространстве точек данных, цветовая раскраска которых была выполнена с учётом результатов агломеративной кластеризации в исходных многомерных пространствах. При этом только для практического набора results_2024.csv заметна отделимость некоторых кластеров (рисунки 8,б, 9,б и 10, б). Для двух других наборов данных – results_2023.csv и results_2025.csv – можно отметить близкое расположение кластеров по отношению друг другу (однако без существенного перемешивания точек данных, принадлежащих разным кластерам). Следует отметить, что число кластеров, равное 5, для всех трёх практических наборов данных выбрано только с целью визуализации (без априорных знаний о фактическом числе кластеров).

4.3. Сравнительный анализ временных затрат на реализацию алгоритма PaCMAP при использовании в его составе различных алгоритмов снижения размерности

При выполнении экспериментальных исследований на тестовых и практических наборах данных была выполнена фиксация оценок временных затрат на реализацию одного запуска алгоритма PaCMAP при использовании в его составе различных алгоритмов снижения размерности.

На рисунке 11 приведена столбчатая диаграмма с оценками временных затрат при реализации одного запуска алгоритма PaCMAP при использовании алгоритма PCA и алгоритмов Isomap, UMAP, TriMAP, так как они хотя бы один раз попали в рейтинг 3 лучших в ходе экспериментальных исследований по сценарию AI/RoP при значениях параметров алгоритмов, заданных по умолчанию.

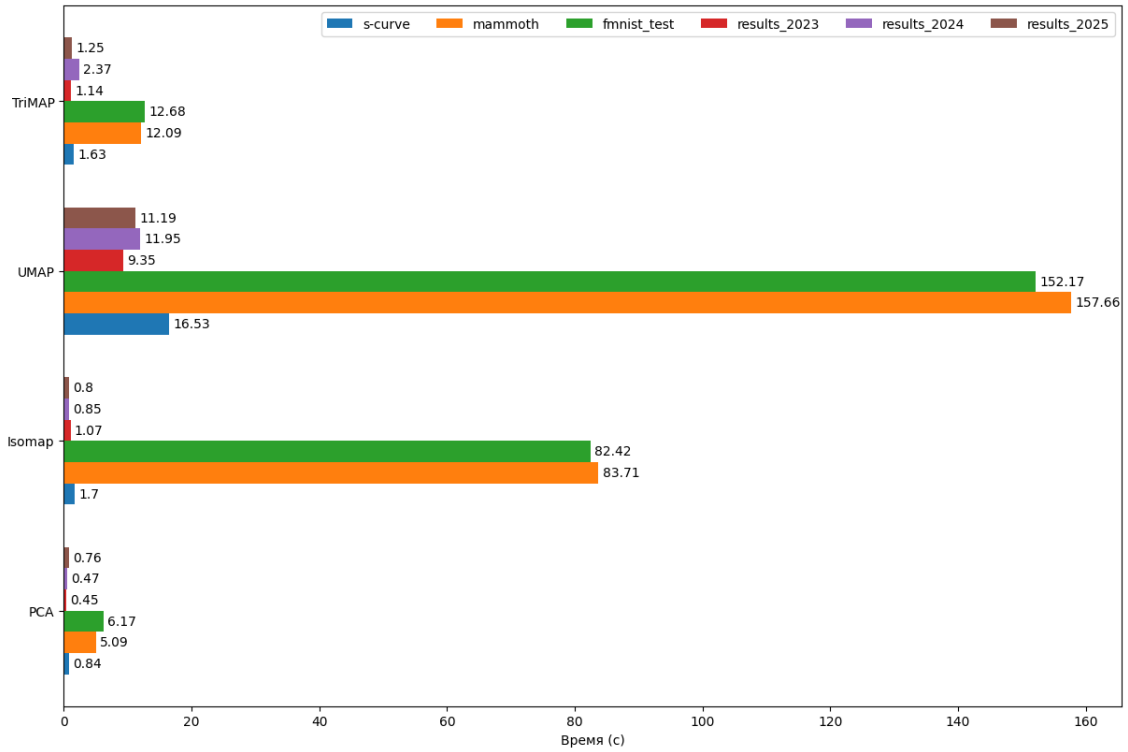


Рисунок 11. Оценка временных затрат для тестовых и практических наборов данных при реализации алгоритма PaCMAP при использовании PCA, Isomap, UMAP, TriMAP снижения размерности по сценарию AI/RoP

На рисунке 12 приведена столбчатая диаграмма с оценками временных затрат при реализации одного запуска алгоритма PaCMAP при использовании алгоритмов MSD, SE, так как они хотя бы один раз попали в рейтинг 3 лучших в ходе экспериментальных исследований по сценарию AI/RoP при значениях параметров алгоритмов, заданных по умолчанию. Графическое представление оценок временных затрат на рисунках 11 и 12 обусловлено желанием наглядно отобразить как малые, так и большие значения оценок временных затрат.

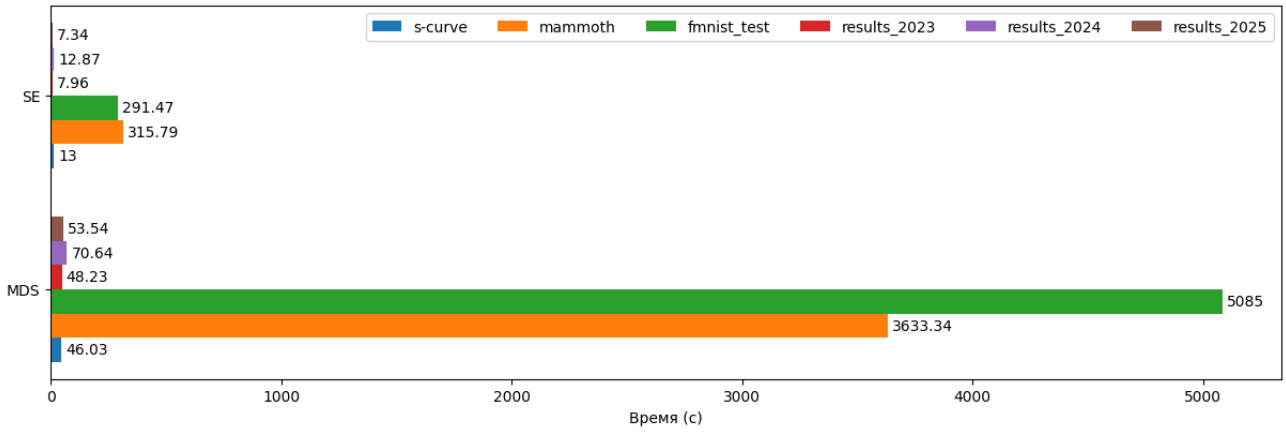


Рисунок 12. Оценка временных затрат для тестовых и практических наборов данных при реализации алгоритма PaCMAP при использовании алгоритмов MSD и SE по сценарию AI/RoP

С точки зрения минимизации временных затрат на получение результатов вложения набора данных в двухмерное пространство можно сделать следующие выводы. Самой времязатратной оказалась реализация запуска алгоритма PaCMAP при использовании алгоритма MDS в его составе. Затем следуют реализации запусков алгоритма PaCMAP при использовании алгоритмов SE, UMAP, Isomap, TriMAP и PCA в его составе, упорядоченных по убыванию оценок временных затрат на реализации запусков.

Очевидно, что использование алгоритма MDS (особенно при работе многомерными наборами данных, содержащими большое число наблюдений) в составе алгоритма PaCMAP сопряжено со значительными временными затратами, которые на 1-3 порядка больше, чем при использовании других алгоритмов снижения размерности (таблица 3).

Использование алгоритма SE в составе алгоритма PaCMAP также сопровождается большими временными затратами, однако они на порядок меньше, чем при использовании алгоритма MDS в составе алгоритма PaCMAP.

При этом оба алгоритма – MSD и SE – были определены в качестве лидеров наряду с алгоритмом UMAP в смысле минимизации функции потерь при их включении в состав алгоритма PaCMAP.

Заключение

Результаты проведенных экспериментальных исследований по использованию 11 алгоритмов снижения размерности в составе алгоритма PaCMAP позволяют сделать выводы о целесообразности включения некоторых из них в состав алгоритма PaCMAP как с учетом полученных оценок значений функции потерь, так и с учетом оценок временных затрат. В большинстве случаев на тестовых и практических наборах данных лидерами (в смысле минимизации функции потерь) среди 11 алгоритмов снижения размерности (таблица 3) при включении их в состав алгоритма PaCMAP оказались алгоритмы MDS, UMAP и SE. При этом использование алгоритмов MDS и SE в составе алгоритма PaCMAP приводит к росту временных затрат на его реализацию. Кроме того, при выполнении экспериментальных исследований на тестовых и практических наборах данных существенную роль играла (в смысле минимизации функции потерь) именно алгоритмическая инициализация вложения данных с применением ряда алгоритмов снижения размерности (таблица 3).

В дальнейшем планируется исследовать целесообразность использования различных алгоритмов оптимизации в составе алгоритма PaCMAP наряду с алгоритмом Adam, как это предложено авторами [15]. Модифицированные версии алгоритма PaCMAP могут быть применены к анализу практических наборов данных, полученных при решении практических задач, в частности, к наборам данных, применяемым для выявления различных типов студенческой активности при онлайн-обучении программированию на языке Python [29, 30] с целью принятия тех или иных, в том числе управленческих и корректирующих, решений.

Список литературы

1. Crawford I.M., Lomas R.A. Factory Analysis – a Tool for Data Reduction, European Journal of Marketing. – 1980. – Vol. 14(7). – P. 414–421. https://doi.org/10.1108/EUM0000000004917.
2. Desai C. Understanding Linear Discriminant Analysis for Classification and Dimensionality Reduction // International Journal of Research and Analytical Reviews. – 2022. – Vol. 9(4). – P. 202–209.
3. Jolliffe I.T., Cadima J. Principal component analysis: a review and recent developments // Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. – 2016. – Vol. 374 (2065). – P. 20150202. https://doi.org/10.1098/rsta.2015.0202.
4. Tenenbaum J.B., de Silva V., Langford J.C. A global geometric framework for nonlinear dimensionality reduction // Science. – 2000. – Vol. 290(5500). – P. 2319–2323.
5. Roweis S.T., Lawrence S.K. Nonlinear dimensionality reduction by locally linear embedding // Science. – 2000. – Vol. 290(5500). – P. 2323–2326.
6. Schölkopf B., Platt J., Hofmann T. MLLE: Modified Locally Linear Embedding Using Multiple Weights // Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference, MIT Press. – 2007. – P. 1593–1600.
7. Donoho D.L., Grimes C. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data // Proceedings of the National Academy of Arts and Sciences. – 2003. – Vol. 100(10). – P. 5591–5596.
8. Zhang Z., Zha H. Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment // SIAM Journal on Scientific Computing. – 2004. – Vol. 26 (1). – P. 313–338. – https://doi.org/10.1137/s1064827502419154.
9. Mead A. Review of the Development of Multidimensional Scaling Methods // Journal of the Royal Statistical Society. Series D (The Statistician). – 1992. – Vol. 41(1). – P. 27–39. https://doi.org/10.2307/2348634.
10. Belkin M., Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering // Advances in Neural Information Processing Systems. – 2001. – Vol.14. – P. 585–591. MIT Press.
11. van der Maaten L., Hinton G. Visualizing data using t-SNE. Journal of Machine Learning Research. – 2008. – Vol. 9. – P. 2579–2605.
12. Tang J., Liu J., Zhang M., Mei Q. Visualizing large-scale and highdimensional data // Proceedings of the 25th International Conference on the World Wide Web. – 2016. – P. 287–297.
13. McInnes L., Healy J., Melville J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction // arXiv:1802.03426. – 2018. https://doi.org/10.48550/arXiv.1802.03426.
14. Amid E., Warmuth M.K. TriMap: Large-scale Dimensionality Reduction Using Triplets // arXiv:1910.00204. – 2019. https://doi.org/10.48550/arXiv.1910.00204.
15. Wang Y., Huang H., Rudin C., Shaposhnik Y. Understanding How Dimension Reduction Tools Work: An Empirical Approach to Deciphering t-SNE, UMAP, TriMap, and PaCMAP for Data Visualization // Journal of Machine Learning Research. – 2021. – Vol. 22(201). – P. 1−73.
16. Becht E., McInnes L., Healy J., Dutertre C.A., Kwok I.W.H., Ng L.G., Ginhoux F., Newell E.W. Dimensionality reduction for visualizing single-cell data using UMAP // Nature Biotechnology. – 2019. – Vol. 37. – P. 38–44.
17. Dorrity M.W., Saunders L.M., Queitsch C., Fields S., Trapnell C. Dimensionality reduction by UMAP to visualize physical and genetic interactions // Nature Communications. – 2020. – Vol. 11. – P. 1537.
18. Demidova L.A. A Novel Approach to Decision-Making on Diagnosing Oncological Diseases Using Machine Learning Classifiers Based on Datasets Combining Known and/or New Generated Features of a Different Nature // Mathematics. – 2023. – Vol. 11. – P. 792. https://doi.org/10.3390/math11040792.
19. Demidova L.A. Decision-Making on the Diagnosis of Oncological Diseases Using Cost-Sensitive SVM Classifiers Based on Datasets with a Variety of Features of Different Natures. – Mathematics. – 2024. – Vol. 12. – P. 538. https://doi.org/10.3390/math12040538.
20. Kingma D.P., Ba J. Adam: A method for stochastic optimization // arXiv:1412.6980. – 2015. https://doi.org/10.48550/arXiv.1412.6980.
21. Vasiliev A.V., Melnikov A.O., Lesko S.A. Robust neural network filtering in the tasks of building intelligent interfaces // Russian Technological Journal. – 2023. – Vol. 11(2). P. 7–19. https://doi.org/10.32362/2500-316X-2023-11-2-7-19.
22. Damrich S., Hamprecht F.A. On UMAP's true loss function // arXiv:2103.14608. – 2021. https://doi.org/10.48550/arXiv.2103.14608.

23. Demidova L.A., Gorchakov A.V. Fuzzy Information Discrimination Measures and Their Application to Low Dimensional Embedding Construction in the UMAP Algorithm // Journal of Imaging. – 2022. – Vol. 8. – P. 113. https://doi.org/10.3390/jimaging8040113.

24. PaCMAP. Available online: https://github.com/YingfanWang/PaCMAP/tree/master (accessed on 12 November 2025).

25. Comparison of Manifold Learning methods. Available online: https://scikit-learn.org/stable/auto_examples/manifold/plot_compare_methods.html#sphx-glr-auto-examples-manifold-plot-compare-methods-py (accessed on 12 November 2025).

26. Mammoth. Available online: https://github.com/MNoichl/UMAP-examples-mammoth-/blob/master/mammoth_a.csv (accessed on 12 November 2025).

27. Fashion MNIST. Available online: https://www.kaggle.com/datasets/zalando-research/fashionmnist (accessed on 12 November 2025).

28. Demidova L.A., Andrianova E.G., Sovietov P.N., Gorchakov A.V. Dataset of Program Source Codes Solving Unique Programming Exercises Generated by Digital Teaching Assistant // Data – 2023. – Vol. 8. – P. 109. https://doi.org/10.3390/data8060109.

29. Demidova L.A., Sovietov P.N, Andrianova E.G., Demidova A.A. Anomaly Detection in Student Activity in Solving Unique Programming Exercises: Motivated Students against Suspicious Ones // Data. – 2023. – Vol. 8. – Vol. 129. https://doi.org/10.3390/data8080129.

30. Andrianova E., Demidov N. An Approach to Identifying Suspicious Student Activities During Online Programming Training Based on One-Class Classifiers // 2023 5th International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA), Lipetsk, Russian Federation. – 2023. P. 288–293. https://doi.org/10.1109/SUMMA60232.2023.10349673.

31. Andrianova E., Demidov N. Study of Approaches to Preprocessing Datasets and Initializationof Embeddings in Low-Dimensional Space when Implementing the PacMAP Algorithm // 2024 6th International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA), Lipetsk, Russian Federation. – 2024. – P. 370–377. https://doi.org/10.1109/SUMMA64428.2024.10803872.

References

1. Crawford I.M., Lomas R.A. Factory Analysis – a Tool for Data Reduction, European Journal of Marketing. – 1980. – Vol. 14(7). – P. 414–421. https://doi.org/10.1108/EUM0000000004917.

2. Desai C. Understanding Linear Discriminant Analysis for Classification and Dimensionality Reduction // International Journal of Research and Analytical Reviews. – 2022. – Vol. 9(4). – P. 202–209.

3. Jolliffe I.T., Cadima J. Principal component analysis: a review and recent developments // Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. – 2016. – Vol. 374 (2065). – P. 20150202. https://doi.org/10.1098/rsta.2015.0202.

4. Tenenbaum J.B., de Silva V., Langford J.C. A global geometric framework for nonlinear dimensionality reduction // Science. – 2000. – Vol. 290(5500). – P. 2319–2323.

5. Roweis S.T., Lawrence S.K. Nonlinear dimensionality reduction by locally linear embedding // Science. – 2000. – Vol. 290(5500). – P. 2323–2326.

6. Schölkopf B., Platt J., Hofmann T. MLLE: Modified Locally Linear Embedding Using Multiple Weights // Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference, MIT Press. – 2007. – P. 1593–1600.

7. Donoho D.L., Grimes C. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data // Proceedings of the National Academy of Arts and Sciences. – 2003. – Vol. 100(10). – P. 5591–5596.

8. Zhang Z., Zha H. Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment // SIAM Journal on Scientific Computing. – 2004. – Vol. 26 (1). – P. 313–338. – https://doi.org/10.1137/s1064827502419154.

9. Mead A. Review of the Development of Multidimensional Scaling Methods // Journal of the Royal Statistical Society. Series D (The Statistician). – 1992. – Vol. 41(1). – P. 27–39. https://doi.org/10.2307/2348634.

10. Belkin M., Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering // Advances in Neural Information Processing Systems. – 2001. – Vol.14. – P. 585–591. MIT Press.

11. van der Maaten L., Hinton G. Visualizing data using t-SNE. Journal of Machine Learning Research. – 2008. – Vol. 9. – P. 2579–2605.

12. Tang J., Liu J., Zhang M., Mei Q. Visualizing large-scale and highdimensional data // Proceedings of the 25th International Conference on the World Wide Web. – 2016. – P. 287–297.

13. McInnes L., Healy J., Melville J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction // arXiv:1802.03426. – 2018. https://doi.org/10.48550/arXiv.1802.03426.

14. Amid E., Warmuth M.K. TriMap: Large-scale Dimensionality Reduction Using Triplets // arXiv:1910.00204. – 2019. https://doi.org/10.48550/arXiv.1910.00204.

15. Wang Y., Huang H., Rudin C., Shaposhnik Y. Understanding How Dimension Reduction Tools Work: An Empirical Approach to Deciphering t-SNE, UMAP, TriMap, and PaCMAP for Data Visualization // Journal of Machine Learning Research. – 2021. – Vol. 22(201). – P. 1−73.

16. Becht E., McInnes L., Healy J., Dutertre C.A., Kwok I.W.H., Ng L.G., Ginhoux F., Newell E.W. Dimensionality reduction for visualizing single-cell data using UMAP // Nature Biotechnology. – 2019. – Vol. 37. – P. 38–44.

17. Dorrity M.W., Saunders L.M., Queitsch C., Fields S., Trapnell C. Dimensionality reduction by UMAP to visualize physical and genetic interactions // Nature Communications. – 2020. – Vol. 11. – P. 1537.

18. Demidova L.A. A Novel Approach to Decision-Making on Diagnosing Oncological Diseases Using Machine Learning Classifiers Based on Datasets Combining Known and/or New Generated Features of a Different Nature // Mathematics. – 2023. – Vol. 11. – P. 792. https://doi.org/10.3390/math11040792.

19. Demidova L.A. Decision-Making on the Diagnosis of Oncological Diseases Using Cost-Sensitive SVM Classifiers Based on Datasets with a Variety of Features of Different Natures. – Mathematics. – 2024. – Vol. 12. – P. 538. https://doi.org/10.3390/math12040538.

20. Kingma D.P., Ba J. Adam: A method for stochastic optimization // arXiv:1412.6980. – 2015. https://doi.org/10.48550/arXiv.1412.6980.

21. Vasiliev A.V., Melnikov A.O., Lesko S.A. Robust neural network filtering in the tasks of building intelligent interfaces // Russian Technological Journal. – 2023. – Vol. 11(2). P. 7–19. https://doi.org/10.32362/2500-316X-2023-11-2-7-19.

22. Damrich S., Hamprecht F.A. On UMAP's true loss function // arXiv:2103.14608. – 2021. https://doi.org/10.48550/arXiv.2103.14608.

23. Demidova L.A., Gorchakov A.V. Fuzzy Information Discrimination Measures and Their Application to Low Dimensional Embedding Construction in the UMAP Algorithm // Journal of Imaging. – 2022. – Vol. 8. – P. 113. https://doi.org/10.3390/jimaging8040113.

24. PaCMAP. Available online: https://github.com/YingfanWang/PaCMAP/tree/master (accessed on 12 November 2025).

25. Comparison of Manifold Learning methods. Available online: https://scikit-learn.org/stable/auto_examples/manifold/plot_compare_methods.html#sphx-glr-auto-examples-manifold-plot-compare-methods-py (accessed on 12 November 2025).

26. Mammoth. Available online: https://github.com/MNoichl/UMAP-examples-mammoth-/blob/master/mammoth_a.csv (accessed on 12 November 2025).

27. Fashion MNIST. Available online: https://www.kaggle.com/datasets/zalando-research/fashionmnist (accessed on 12 November 2025).

28. Demidova L.A., Andrianova E.G., Sovietov P.N., Gorchakov A.V. Dataset of Program Source Codes Solving Unique Programming Exercises Generated by Digital Teaching Assistant // Data – 2023. – Vol. 8. – P. 109. https://doi.org/10.3390/data8060109.

29. Demidova L.A., Sovietov P.N, Andrianova E.G., Demidova A.A. Anomaly Detection in Student Activity in Solving Unique Programming Exercises: Motivated Students against Suspicious Ones // Data. – 2023. – Vol. 8. – Vol. 129. https://doi.org/10.3390/data8080129.

30. Andrianova E., Demidov N. An Approach to Identifying Suspicious Student Activities During Online Programming Training Based on One-Class Classifiers // 2023 5th International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA), Lipetsk, Russian Federation. – 2023. P. 288–293. https://doi.org/10.1109/SUMMA60232.2023.10349673.

31. Andrianova E., Demidov N. Study of Approaches to Preprocessing Datasets and Initializationof Embeddings in Low-Dimensional Space when Implementing the PacMAP Algorithm // 2024 6th International Conference on

Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA), Lipetsk, Russian Federation. – 2024. – P. 370–377. https://doi.org/10.1109/SUMMA64428.2024.10803872.