

УЛУЧШЕНИЕ КАЧЕСТВА ПРОГНОЗОВ МОДЕЛИ EXTREME LEARNING MACHINE НА ЗАДАЧАХ РЕГРЕССИИ И КЛАССИФИКАЦИИ С ПОМОЩЬЮ СЕЛЕКЦИИ ФУНКЦИЙ АКТИВАЦИИ ПОСРЕДСТВОМ АЛГОРИТМА GENE EXPRESSION PROGRAMMING

Демидова Л.А., Журавлев В.Е.

Федеральное государственное бюджетное образовательное учреждение высшего образования «МИРЭА — Российский технологический университет», 119454, Россия, г. Москва, проспект Вернадского, 78, e-mail: liliya.demidova@rambler.ru, vovcrane@mail.ru

В данной работе рассматривается подход к улучшению качества прогнозирования Extreme Learning Machine (ELM) с помощью генетического алгоритма, реализующего процесс эволюции для функций активации нейронов в скрытом слое модели. Для выбора наилучшего кандидата в процессе селекции учитывается как его вычислительная сложность, так и качество результатов, получаемых с его использованием. Для апробации предложенного подхода рассматривается 4 разных набора данных, используемых для оценки качества различных моделей при решении задач классификации и регрессии. Результаты экспериментальных исследований подтверждают, что с функциями активации, полученными с помощью генетического алгоритма Gene Expression Programming (GEP), модель ELM показывает более качественные результаты, чем с применением классических функций активации, используемых для решения подобных задач.

Ключевые слова: машинное обучение, классификация, регрессия, символьная регрессия, Extreme Learning Machine (ELM), Gene Expression Programming (GEP), генетические алгоритмы, статистический критерий Манна-Уитни.

IMPROVING THE QUALITY OF EXTREME LEARNING MACHINE PREDICTIONS ON REGRESSION AND CLASSIFICATION TASKS BY EMPLOYING SELECTION BREEDING OF ACTIVATION FUNCTIONS VIA THE GENE EXPRESSION PROGRAMMING ALGORITHM

Demidova L.A., Zhuravlev V.E.

Federal State Budgetary Educational Institution of Higher Education "MIREA - Russian Technological University", 119454, Russia, Moscow, Vernadsky Avenue, 78, e-mail: liliya.demidova@rambler.ru, vovcrane@mail.ru.

This paper proposes an approach to improve the quality of Extreme Learning Machine (ELM) predictions using a genetic algorithm that implements an evolutionary process for neuron activation functions in the hidden layer of the model. To select the best candidate in the selection breeding process, both its computational complexity and the quality of the results obtained using it are considered. To validate the proposed approach, 4 different datasets are considered and used to evaluate the quality of different models for classification and regression tasks. Experimental results confirm that ELM model shows better results with activation functions obtained using Gene Expression Programming (GEP) algorithm than with classical activation functions used to solve similar problems.

Keywords: machine learning, classification, regression, symbolic regression, Extreme Learning Machine (ELM), Gene Expression Programming (GEP), genetic algorithms, Mann-Whitney U test.

Введение

С постоянно увеличивающимися во всем мире объемами информации растет также и необходимость в их анализе и обработке. Для этого используются разные методы и алгоритмы, выбор которых зависит от формата

данных, сферы их применения и многих других факторов. Однако для обработки информации со сложной и нечеткой структурой уже недостаточно классических алгоритмических подходов, основанных на четких правилах, поэтому для таких целей активно применяются интеллектуальные инструменты и технологии, использующие машинное обучение и искусственные нейронные сети. Наиболее типовыми проблемами, для решения которых применяются такие инструменты и технологии, являются задачи классификации и регрессии.

Решение задачи регрессии заключается в предсказании непрерывных значений на основании каких-либо признаков. Например, к задачам регрессии относятся задачи прогнозирования стоимости жилья по его характеристикам [1] или интенсивности движения транспортных средств на перекрестке [2].

Решение задачи классификации, в отличие от регрессии, заключается в предсказании дискретных значений, которыми могут служить какие-либо заранее установленные категории или классы объектов. Например, к задачам классификации относятся задачи определения сорта риса по характеристикам, полученным с его фотографии [3], или выбор жанра для книги по ее содержанию [4].

Одной из наиболее известных моделей, успешно решающей задачи классификации и регрессии, а также кластеризации, аппроксимации и т.д., является Extreme Learning Machine (ELM) [5]. ELM представляет собой полносвязную искусственную нейронную сеть, включающую в себя вход, выход и только один скрытый слой. Связи между входом и скрытым слоем, а также смещения его нейронов инициализируются случайными числами при создании модели, а значения весов между скрытым слоем и выходом вычисляются аналитически с использованием обучающей выборки. Преимущество ELM над другими схожими моделями заключается в высокой скорости обучения и сравнительно хорошей производительности.

Тем не менее качество решения задач с помощью ELM, измеряемое выбранными специально под их условия метриками, зависит как от начальной инициализации входных весов модели, так и от используемой функции активации в скрытом слое. По этой причине имеет смысл исследовать способы улучшения качества работы ELM с помощью интеллектуальных алгоритмов подбора значений для ее необучаемых параметров. Например, в работе [6] показано, как использование популяционных алгоритмов оптимизации для подбора значений весов в ELM может статистически значимо увеличить качество ее прогнозов на задачах регрессии.

В данной работе предлагается альтернативный подход к улучшению качества работы ELM, измеряемой такими метриками, как F1-мера и MAPE для классификации и регрессии соответственно, с помощью конструирования наиболее подходящих для конкретных задач функций активации скрытого слоя. Для этой цели будет использоваться Gene Expression Programming (GEP) [7], который представляет собой генетический алгоритм, осуществляющий эволюцию древовидных структур, в качестве которых могут рассматриваться, в том числе, функции активации.

Extreme Learning Machine

Как уже было сказано ранее, ELM представляет собой полносвязную нейронную сеть, включающую в себя только один скрытый слой. Общее устройство данной модели представлено на рисунке 1.

Для обучения ELM необходим как набор входных данных, состоящий из N объектов по n признаков, $X_L \in \mathbb{R}^{N \times n}$, так и набор выходных данных, состоящий из N объектов по m значений, $Y_L \in \mathbb{R}^{N \times m}$. При создании модели матрица весов $W \in \mathbb{R}^{n \times \tilde{N}}$ между n нейронами входного слоя и \tilde{N} нейронами скрытого слоя, а также вектор смещений $\tilde{b} \in \mathbb{R}^{\tilde{N}}$ инициализируются случайными числами и не изменяются в процессе дальнейшего обучения. В качестве функции активации $g(x)$ может использоваться любая нелинейная функция, но обычно применяется сигмоида (1), гиперболический тангенс (2) или ReLU (3):

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}}, \quad (1)$$

$$\text{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad (2)$$

$$\text{ReLU}(x) = \max(0, x). \quad (3)$$

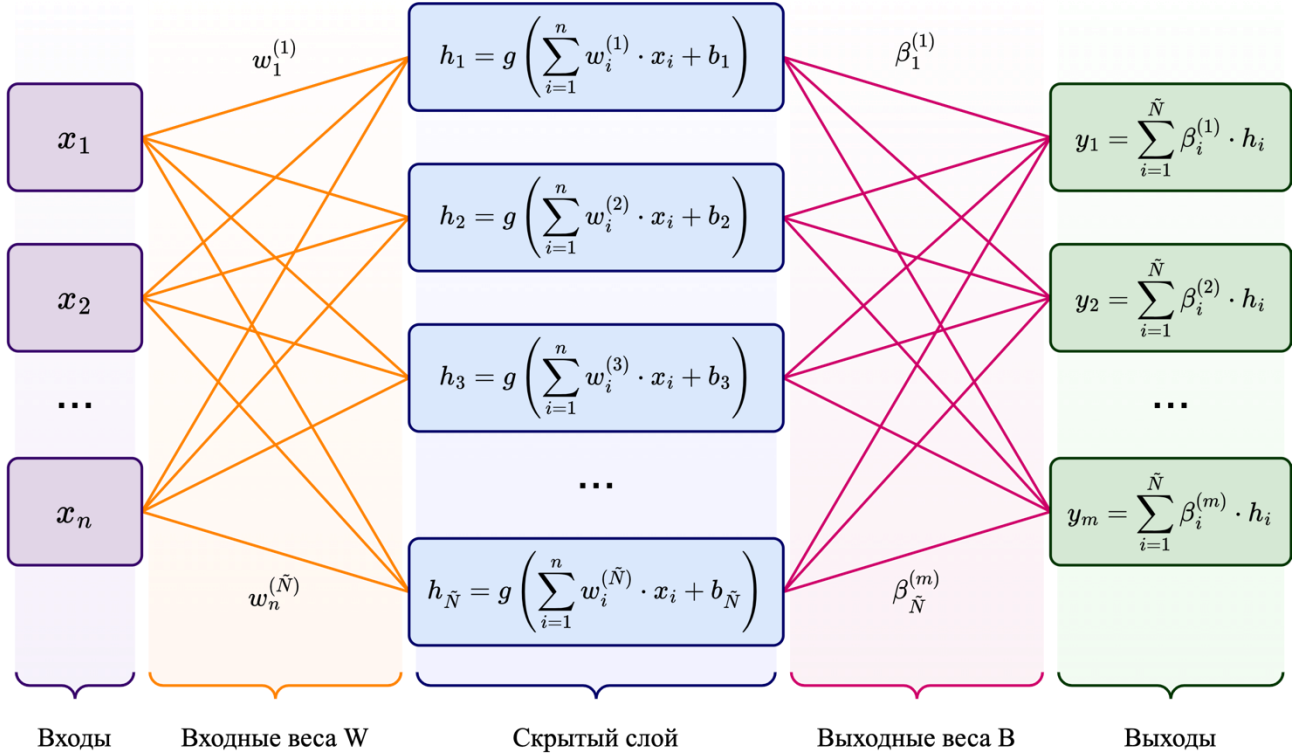


Рисунок 1 – Принцип работы и устройство модели ELM

Как показано на рисунке 1, каждый нейрон скрытого слоя вычисляет взвешенную сумму входов с применением функции активации. Таким образом, на выходе из данного слоя формируется матрица $H \in \mathbb{R}^{N \times \tilde{N}}$:

$$H = g(X_L W + \bar{b}^T). \quad (4)$$

Главная задача обучения ELM состоит в том, чтобы вычислить матрицу весов $B \in \mathbb{R}^{\tilde{N} \times m}$ между скрытым слоем и выходом. Зная, что в результате обработки X_L должна получиться матрица Y_L , можно составить следующее уравнение:

$$HB = Y_L. \quad (5)$$

Для нахождения матрицы выходных весов B необходимо получить $H^+ Y_L$, где H^+ означает псевдообратную матрицу от H , вычисленную по методу Мура – Пенроуза [8].

Расчет предсказаний Y_{pred} для валидационного набора данных X_T выполняется как:

$$Y_{pred} = g(X_T W + \bar{b}^T) B. \quad (6)$$

В зависимости от типа решаемой задачи и особенностей данных, может быть выбрана какая-либо конкретная метрика для оценки производительности ELM, но, в отличие от классического градиентного спуска, используемого для обучения полносвязных нейронных сетей прямого распространения, выбранная метрика будет выполнять лишь функцию оценки качества модели, не участвуя напрямую в настройке ее параметров.

Gene Expression Programming

Gene Expression Programming (программирование экспрессии генов, GEP) – это эволюционный алгоритм, способный производить компьютерные программы или другие модели, представляющие собой сложные древовидные структуры. В процессе эволюции они обучаются и адаптируются, изменяя свою форму и состав, подобно живым организмам. Как и в реальной жизни, производимые GEP модели закодированы в хромосомах, состоящих из генов, образуя тем самым генотип. В процессе его расшифровки (экспрессии) образуются реальные модели, обладающие собственными характеристиками и свойствами. Таким образом, GEP реализует генотип-

фенотипную систему, сочетающую в себе удобство хранения и обработки генетической информации и сложность конечных продуктов эволюции.

Ключевой особенностью GEP является неизменяемость структуры хромосом. В процессе эволюции различные особи скрещиваются друг с другом и мутируют, меняя свой состав, но длина генов и хромосом, которыми они представлены, остается неизменной. Несмотря на это, в процессе экспрессии генов могут получаться древовидные структуры разных форм и размеров.

Каждый ген в GEP состоит из головы (head) и хвоста (tail). Голова содержит как элементы, представляющие разные функции из множества F , так и терминалы из набора T . Хвост, в свою очередь, включает в себя только терминальные элементы. Для определения полной структуры гена необходимо объявить множества F и T , а также длину головной части l_{head} . Длина хвоста l_{tail} определяется как:

$$l_{tail} = l_{head}(a - 1) + 1, \quad (7)$$

где a – количество аргументов у функции из набора F , обладающей максимальной арифметичностью.

Хромосома в GEP состоит из заранее определенного количества генов, поэтому также обладает неизменной длиной. В процессе экспрессии каждый ген формирует собственное дерево. Деревья могут объединяться в одну конструкцию с помощью специальной функции (как правило, при эволюции деревьев выражений для таких целей используется функция, реализующая операцию сложения). На рисунке 2 представлен пример процесса расшифровки хромосомы, состоящей из двух генов.

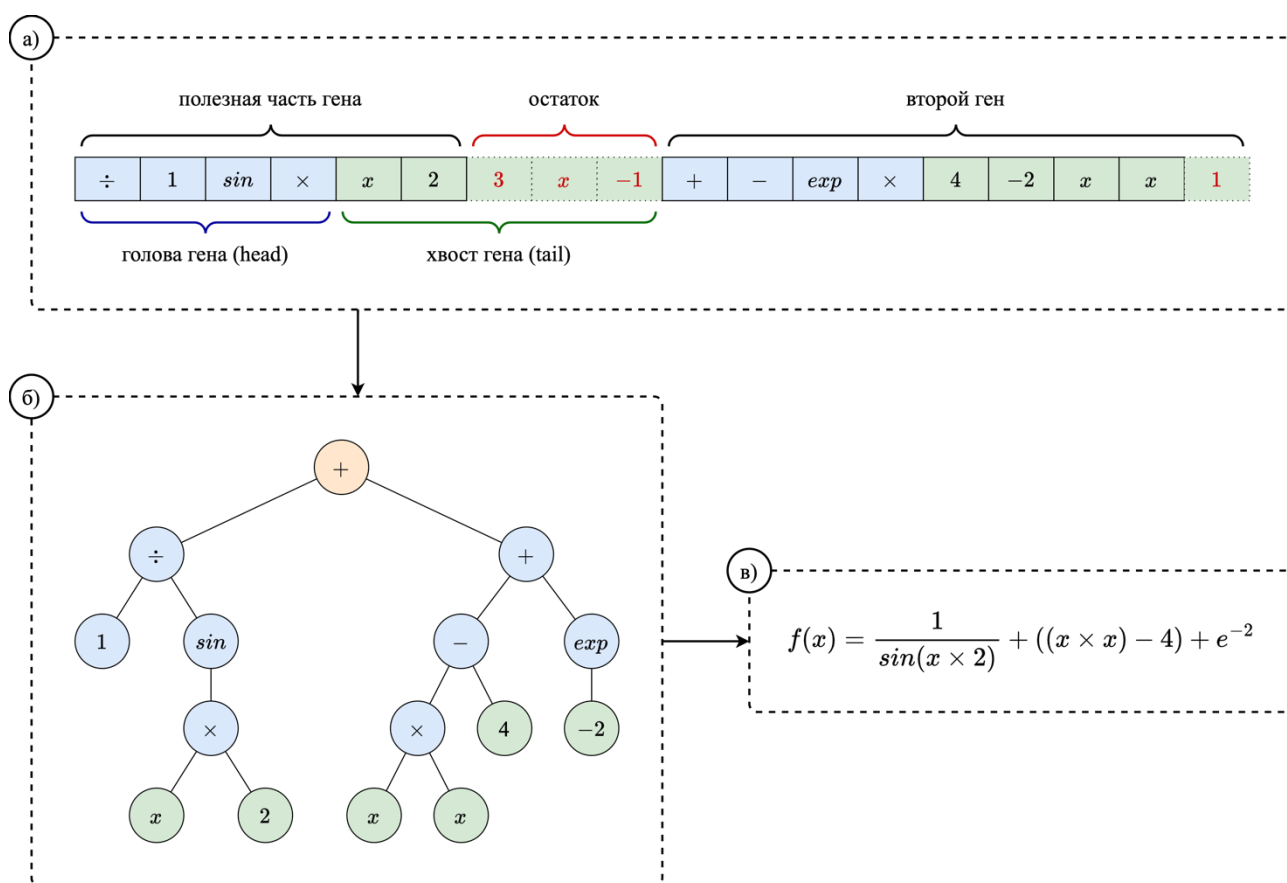


Рисунок 2 – Процесс экспрессии генов в GEP: а – исходная хромосома, состоящая из двух генов, б – построенное по генотипу дерево выражения, в – итоговая функция, полученная из дерева выражения

Как видно из рисунка 2, некоторые фрагменты хромосомы (отмеченные на рисунке 2,а красным цветом) не стали частью итогового дерева выражения. Это связано с тем, что процесс преобразования гена в древовидную структуру осуществляется методом обхода в ширину, как это показано на рисунке 3.

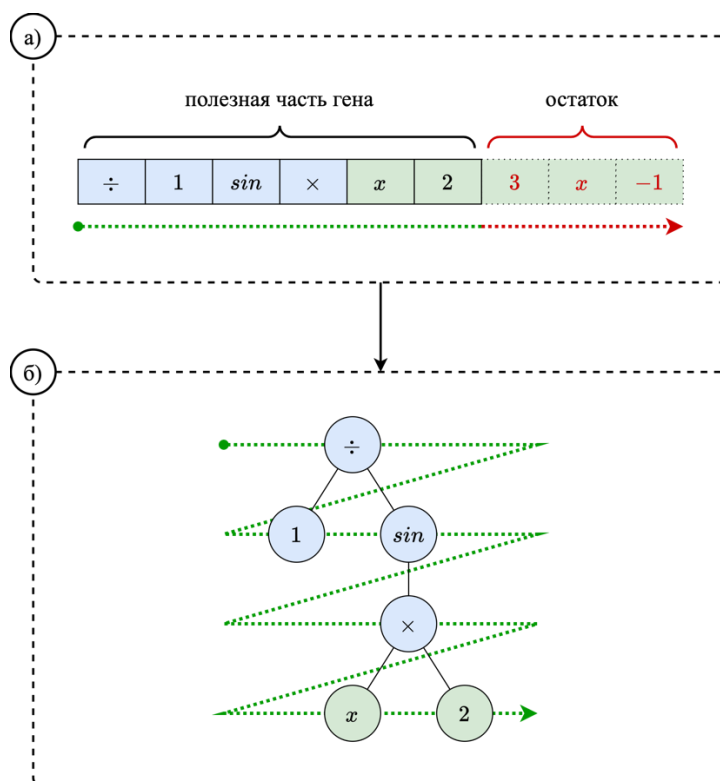


Рисунок 3 – Пример расшифровки гена (а) в древовидную структуру (б) с использованием обхода в ширину (по зеленой пунктирной стрелке)

Такая особенность процесса экспрессии генов в GEP позволяет генерировать разнообразные по форме и содержанию древовидные структуры, используя фиксированный размер хромосом. При этом нет необходимости в проверке полученных деревьев на корректность, поскольку обход в ширину с учетом ариности функций исключает ошибки при их построении.

Сам процесс эволюции хромосом в GEP четко не определен, поэтому может быть настроен индивидуально для конкретной решаемой задачи. В общем случае он включает в себя формирование начальной популяции, агенты которой на протяжении определенного числа поколений скрещиваются друг с другом, мутируют и отбираются, оптимизируя какую-либо заранее составленную функцию потерь, оценивающую приспособленность кандидатов.

Эволюция функций активации

Предлагаемый подход к повышению качества работы ELM состоит в подборе наиболее приспособленной функции активации для скрытого слоя с помощью генетического алгоритма GEP. При этом приспособленность кандидата определяется не только качеством работы модели с ним, но и его вычислительной сложностью. Для этого предлагается рассматривать функции активации как деревья выражений, эволюция которых приведет к формированию наиболее подходящей модели для решения конкретной задачи на классификацию или регрессию.

Чем больше нейронов в скрытом слое имеет ELM, тем более сложные закономерности модель способна анализировать. Однако вместе с этим повышается и вероятность ее переобучения, особенно на небольших объемах данных. С другой стороны, маленькое количество нейронов в скрытом слое ухудшает аппроксимирующие свойства модели, что может привести к ее неспособности распознавания закономерностей в данных. Поэтому, учитывая высокую скорость обучения ELM, имеет смысл заранее протестировать разные размеры скрытого слоя и выбрать тот, на котором модель в среднем работает лучше всего.

Для оценки сложности функций активации предлагается измерять среднее время, которое требуется на их вычисление. Такой простой метод имеет место в данном контексте, поскольку главной причиной оптимизации вычислительной сложности функций в процессе их эволюции является ускорение работы ELM. Таким образом, для измерения среднего времени вычисления функции активации $g(x)$ используется следующая формула:

$$\text{time}(g(x)) = \frac{\sum_{i=1}^{Q_0} \text{gauge}(g(x))}{Q_0}, \quad (8)$$

где $\text{gauge}(g(x))$ – функция, засекающая время вычисления функции $g(x)$ в миллисекундах, применяя ее к каждому значению многомерного массива $x \in \mathbb{R}^{s_1 \times s_2 \times \dots \times s_p}$, состоящего из случайных чисел, взятых из диапазона $[-10; 10]$, а Q_0 – количество замеров для усреднения.

В целях стабилизации замеров времени расчеты по формуле (8) выполняются Q_1 раз, после чего находится медиана $\text{Me}(g(x))$ из Q_1 полученных значений:

$$\text{Me}(g(x)) = \text{median}(\text{time}_1(g(x)), \text{time}_2(g(x)), \dots, \text{time}_{Q_1}(g(x))). \quad (9)$$

Итоговое значение метрики, оценивающей сложность функции активации $g(x)$, вычисляется по формуле:

$$\text{complexity}(g(x)) = -\log_{10}(\text{Me}(g(x))). \quad (10)$$

Логарифмирование в данном случае повышает интерпретируемость метрики, представляя сложность функций активации в виде небольшого положительного числа. Чем меньше значение $\text{complexity}(g(x))$, тем более сложной является функция активации $g(x)$.

Для оценки качества решения задач классификации и регрессии, достигаемых моделью с использованием функции активации $g(x)$, модель ELM d раз инициализируется случайными входными весами, обучается и тестируется с помощью k -проходной перекрестной проверки (k -fold cross-validation). Тестирование осуществляется с использованием специфической для решаемой задачи метрики качества, которой может быть точность, полнота, F1-мера, среднеквадратичная ошибка и т.д. В итоге на основе d результатов вычисляется медиана, которая и служит показателем качества работы ELM $\text{quality}(g(x))$ с функцией активации $g(x)$ в контексте конкретной решаемой задачи.

Для объединения показателя сложности функции $\text{complexity}(g(x))$ и качества ее работы $\text{quality}(g(x))$ используется формула вида:

$$\text{score}(g(x)) = \text{quality}(g(x)) \pm \text{penalty} \times \text{complexity}(g(x)), \quad (11)$$

где penalty представляет собой степень учета сложности функции активации в общей оценке, а знак перед ним зависит от решаемой задачи: при максимизации $\text{quality}(g(x))$ используется знак плюс, а при минимизации – минус.

В дальнейших численных экспериментах для описанных в данном разделе параметров были выбраны значения, перечисленные в таблице 1.

Таблица 1 – Значения параметров для оценки производительности и качества прогнозирования ELM

параметр	Q_0	p	(s_1, s_2, \dots, s_p)	Q_1	d	k	penalty
значение	250	3	(8, 8, 8)	25	10	10	0,01

При решении задач регрессии рекомендуется для вычисления $\text{quality}(g(x))$ использовать такие метрики качества, как MSPE, MAPE и SMAPE, поскольку их относительность позволяет применять одно и то же значение параметра penalty для разных наборов данных. Тем не менее можно использовать и другие метрики, оперирующие абсолютными величинами, но в таком случае необходимо подбирать значение penalty индивидуально для каждой задачи.

Проведение экспериментов

Апробация предлагаемого подхода для улучшения качества прогнозов модели ELM была выполнена с применением языка программирования Python 3.12 в среде Jupyter Notebook. В ходе экспериментов использовался следующий компьютер: MacBook Pro 13 2017 A1708 (процессор: Intel(R) Core(TM) i5-7360U CPU 2.3 ГГц, 2.3 ГГц, 2 ядра; оперативная память: 8 Гб; 64-разрядная операционная система).

Для проведения эксперимента были выбраны четыре классических набора данных, два из которых подразумевают решение задачи классификации, а другие два – регрессии. Характеристики наборов данных представлены в таблице 2.

Таблица 2 – Наборы данных для экспериментов

Название	Описание	Задача	Количество сущностей	Количество признаков	Количество классов
Concrete Compressive Strength [9]	Содержит разные физические характеристики бетона, на основании которых требуется определить его прочность	Регрессия	1030	8	—
Boston Housing [10]	Содержит данные об окружении дома, требуется предсказать его стоимость	Регрессия	506	13	—
Rice (Cammeo and Osmancik) [11]	На основании фотографий риса двух сортов было сформировано 7 признаков, по которым необходимо выполнить классификацию	Классификация	3810	7	2
Yeast [12]	Требуется предсказать участки клеточной локализации белков, опираясь на их характеристики	Классификация	1484	8	10

Для оценки качества решения задач классификации была выбрана такая метрика, как F1-мера. Для задач регрессии было решено использовать среднюю абсолютную ошибку в процентах (Mean Absolute Percentage Error, MAPE), расчет которой производится по формуле (12):

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right|, \quad (12)$$

где N – количество объектов в выборке, y_i – правильное предсказание для i -того объекта, \hat{y}_i – предсказание, сделанное моделью ELM для i -того объекта.

Выбор метрики MAPE обусловлен тем, что её относительность избавляет от необходимости подстраивать параметр penalty для каждой конкретной задачи, а также повышает интерпретируемость результата, чем выгодно отличается от, например, среднеквадратичной ошибки.

Для улучшения качества классификации и регрессии каждый обучающий набор данных X_L , получаемый в ходе 10-проходной перекрестной проверки, масштабируется таким образом, чтобы среднее значение каждого признака равнялось нулю, а стандартное отклонение – единице.

На рисунке 4 представлены графики зависимости качества работы модели ELM с сигмоидальной функцией активации от количества нейронов в ее скрытом слое для всех рассматриваемых наборов данных.

Учитывая, что в данном эксперименте решение задач регрессии подразумевает минимизацию метрики MAPE, а классификации – максимизацию F1-меры, размер скрытого слоя модели ELM для каждого набора данных был выбран в соответствии с наилучшим образом оптимизирующими используемые метрики минимальными и максимальными медианными значениями, полученными из графиков, изображенных на рисунке 4. Результаты выбора представлены в таблице 3.

Таблица 3 – Размер скрытого слоя модели ELM для каждого набора данных

Набор данных	Размер скрытого слоя
Concrete Compressive Strength	190
Boston Housing	120
Rice (Cammeo and Osmancik)	100
Yeast	90

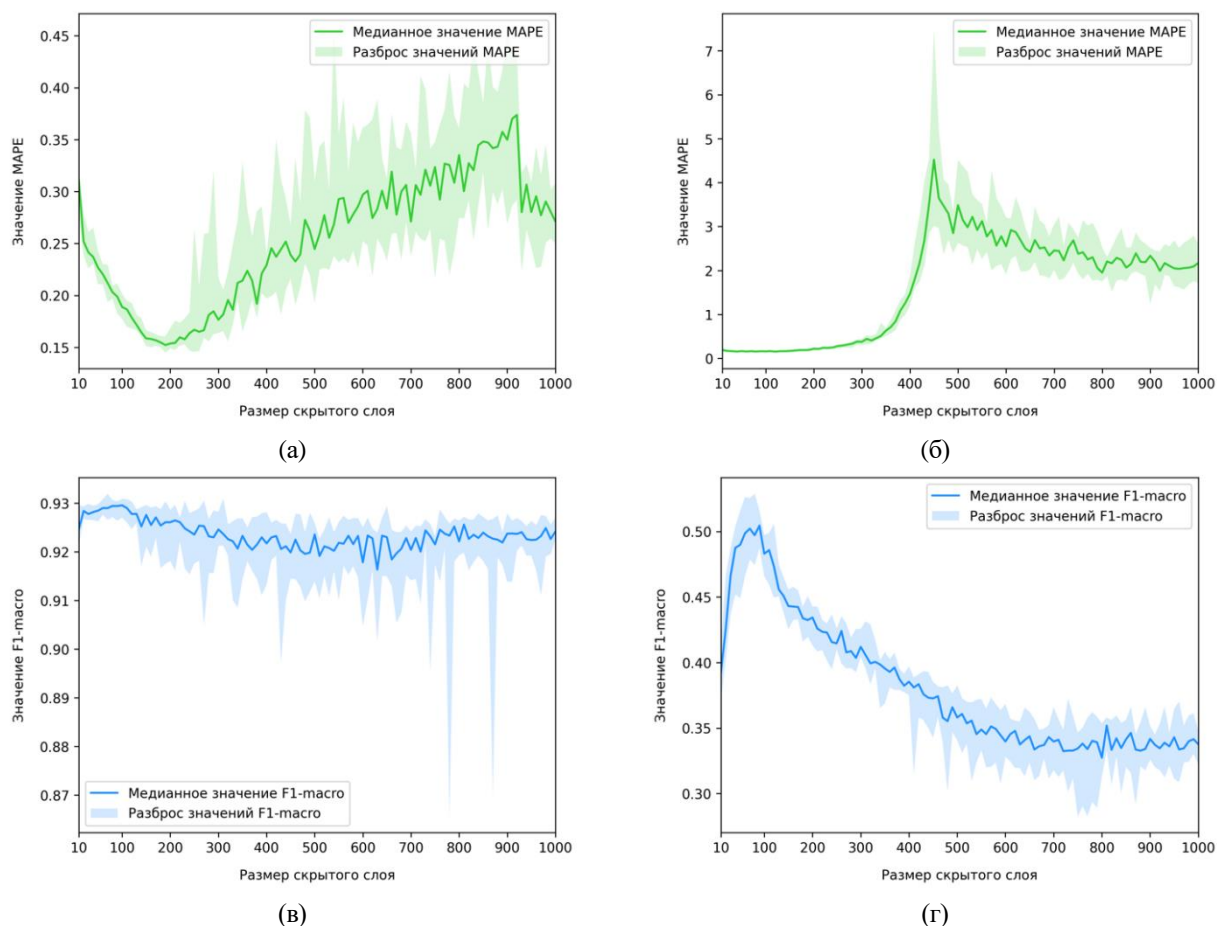


Рисунок 4 – Зависимость качества прогнозов модели ELM от размера скрытого слоя: а – Concrete Compressive Strength, б – Boston Housing, в – Rice (Cammeo and Osmancik), г – Yeast

Для реализации GEP использовалась библиотека `gerppu` [13], предоставляющая все необходимые возможности для проведения экспериментов. Множество функций F , из элементов которого строятся гены, показано в таблице 4.

Таблица 4 – Множество элементарных функций для генов GEP

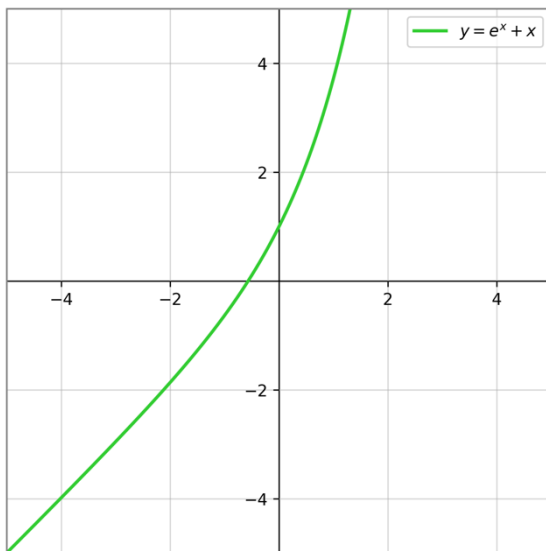
Операция	Арность	Примечание
Сложение	2	—
Вычитание	2	—
Умножение	2	—
Деление	2	Для избежания деления на ноль результатом функции будет делимое, если делитель по модулю меньше, чем $1E-6$
Синус	1	—
Косинус	1	—
Тангенс	1	Если $ \cos(x) < 1E-6$, функция возвращает $\sin(x)$, иначе – $\text{tg}(x)$
Натуральный логарифм	1	$\ln(x)$ не определен при $x < 0$, а вблизи точки 0 стремится к бесконечности, поэтому функция возвращает 0, если аргумент отрицательный или не превышает $1E-6$
Экспонента	1	—
Умножение на (-1)	1	—
Квадратный корень	1	Поскольку квадратный корень не определен для отрицательных чисел, вычисляется модуль от его аргумента
Модуль	1	—
Сигмоида	1	—
Гиперболический тангенс	1	—

Операция	Ариость	Примечание
Сигнум	1	—
Минимум	2	—
Максимум	2	—

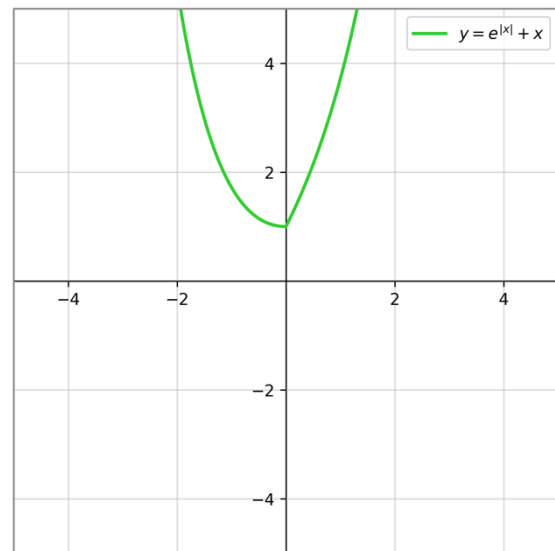
В результате запуска GEP с параметрами, указанными в таблице 5, были получены функции активации, показанные на рисунке 5.

Таблица 5 – Параметры запуска алгоритма GEP

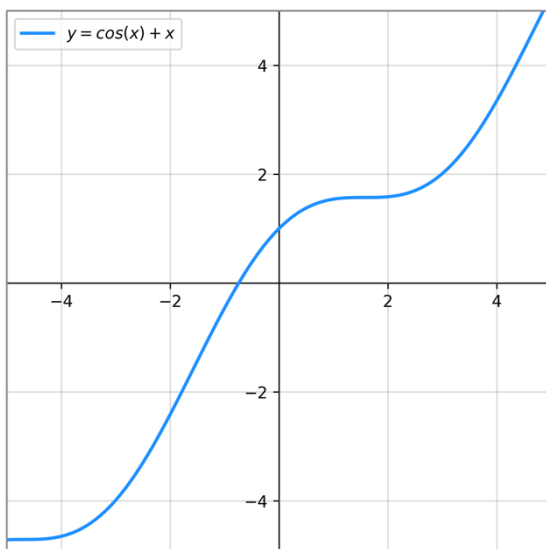
Размер популяции	50
Количество эпох	10
Размер головной части гена	31
Количество генов в хромосоме	2
Механизм отбора	турнир из трех кандидатов
Диапазон значений для случайных констант	от -5 до 5



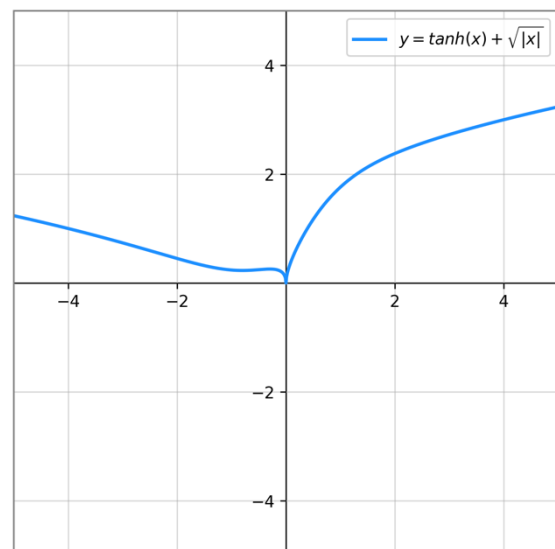
(а)



(б)



(в)



(г)

Рисунок 5 – Функции активации, полученные с помощью GEP: а – Concrete Compressive Strength, б – Boston Housing, в – Rice (Cammeo and Osmancik), г – Yeast

На рисунке 6 показано сравнение полученных функций активации с классическими функциями, часто используемыми в ELM, по показателям метрик качества решаемых задач классификации и регрессии.

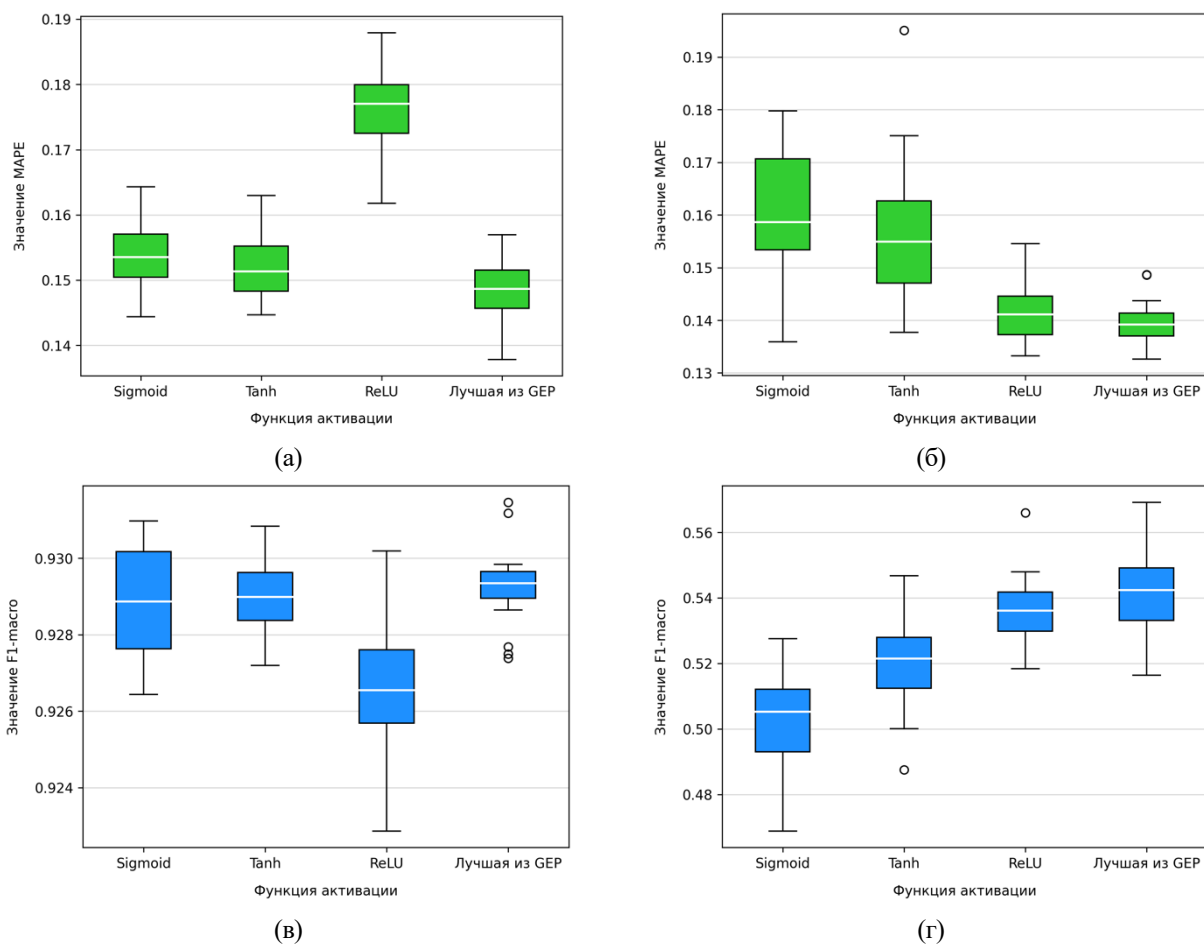


Рисунок 6 – Сравнение полученных функций активации с классическими в виде коробчатых диаграмм: а – Concrete Compressive Strength, б – Boston Housing, в – Rice (Cammeo and Osmancik), г – Yeast

Как видно по результатам сравнения, полученные с помощью GEP функции активации как минимум не уступают своим классическим аналогам, а на некоторых задачах даже превосходят их. Например, на наборе данных Concrete Compressive Strength (рисунок 6,а) медиана значений MAPE для модели ELM, использующей полученную в ходе эволюционного алгоритма функцию активации, находится на одном уровне с нижним квартилем коробчатой диаграммы гиперболического тангенса, показавшего наилучший результат среди классических функций активации. Учитывая факт минимизации метрики качества при решении данной задачи, можно сделать вывод о преимуществе полученной с помощью GEP функции активации над классическими аналогами. Кроме того, по наборам данных Boston Housing и Rice (Cammeo and Osmancik) (рисунки 6,б и 6,в соответственно) видно, что межквартильные расстояния у диаграмм, описывающих полученные с помощью GEP функции, заметно меньше, чем у остальных, что говорит о большей стабильности получаемых с их помощью результатов.

Для более строгого сравнения был применен статистический критерий Манна-Уитни с допустимым уровнем значимости $\alpha = 0,05$. Результаты представлены в таблице 6.

Таблица 6 – Сравнение функций активации с помощью статистического критерия Манна-Уитни

Функция активации	Набор данных			
	Concrete Compressive Strength	Boston Housing	Rice (Cammeo and Osmancik)	Yeast
Sigmoid	+	+	=	+
Tanh	+	+	=	+
ReLU	+	=	+	+

В таблице выше знаком «+» обозначено статистически значимое превосходство функции активации, полученной с помощью GEP, а знаком «=» – отсутствие статистически значимых ситуаций. При этом не было зафиксировано ни одного случая существенного преимущества какой-либо классической функции активации над функцией, полученной в ходе эволюционного алгоритма.

Заключение

По результатам проведенного исследования можно сделать вывод, что использование генетического алгоритма GEP для генерации функций активации скрытого слоя ELM способно повысить качество прогнозов модели на задачах классификации и регрессии. Однако нельзя не отметить, что процесс эволюции требует дополнительных временных затрат, поэтому при решении задач с большими наборами данных, состоящих из множества признаков, возможно, прирост качества работы ELM будет несопоставим с объемом потраченного времени на его достижение. Поэтому в таком случае больший смысл имеет использование одной из классических функций активации, поскольку, согласно проведенным экспериментам, их результативность на задачах классификации и регрессии не так сильно отличается от продуктов эволюции GEP.

Тем не менее в реальных задачах, где в первую очередь важна точность результатов, подход, представленный в данной работе, может оказать существенное влияние на процесс их решения.

Дальнейшие исследования по рассмотренной теме могут быть направлены в сторону модернизации и расширения предложенного в данной работе подхода. Имеет смысл выполнить тестирование разных стратегий отбора и параметров генетического алгоритма, а также дополнить набор элементарных функций, из которых строятся гены. Кроме того, целесообразно исследовать перспективность совмещения предложенного в данной работе подхода с подходом, предполагающим использование популяционных алгоритмов оптимизации для подбора значений входных весов, как это было сделано в [6].

Данная работа была выполнена при поддержке Министерства науки и высшего образования Российской Федерации, тема № FSFZ-2024-0023 «Методы выявления и обработки инцидентов в программном обеспечении на основе интеллектуального анализа мультимодальных векторных представлений текстов программ с применением технологий машинного и глубокого обучения».

Список литературы

1. Гимадиева, Л. Ш. Прогнозирование стоимости жилья / Л. Ш. Гимадиева // Региональные проблемы преобразования экономики. – 2022. – № 3(137). – С. 22-32. – DOI 10.26726/1812-7096-2022-3-22-32.
2. Болодурина, И. П. Интеллектуальная модель прогнозирования интенсивности движения транспортных средств на перекрестке / И. П. Болодурина, Л. М. Анциферова, Л. С. Гришина // Интеллект. Инновации. Инвестиции. – 2022. – № 6. – С. 69-78. – DOI 10.25198/2077-7175-2022-6-69.
3. U. Ilhan. Classification of Osmancik and Cammeo Rice Varieties using Deep Neural Networks / U. Ilhan, A. Ilhan, K. Uyar, E. I. Iseri // 2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) – Ankara, Turkey, 2021. – P. 587-590. – DOI 10.1109/ISMSIT52890.2021.9604606.
4. Николаев, П. Л. Классификация книг по жанрам на основе текстовых описаний посредством глубокого обучения / П. Л. Николаев // International Journal of Open Information Technologies. – 2022. – № 10(1). – С. 36-40.
5. Guang-Bin Huang. Extreme learning machine: Theory and applications / Guang-Bin Huang, Qin-Yu Zhu, Chee-Kheong Siew // Neurocomputing. – 2006. – Vol. 70. – P. 489-501. – DOI 10.1016/j.neucom.2005.12.126.
6. Demidova, L.A. Biology-inspired optimization algorithms applied to intelligent input weights selection of an extreme learning machine in regression problems / L. A. Demidova, A. V. Gorchakov // AIP Conference Proceedings – 2023. – Vol. 2700(1). – DOI 10.1063/5.0124917.
7. Candida Ferreira. Gene Expression Programming: A New Adaptive Algorithm for Solving Problems / Candida Ferreira // Complex Syst. – 2001. – Vol. 13. – DOI 10.48550/arXiv.cs/0102027.
8. Penrose R. On best approximate solutions of linear matrix equations / Penrose R. // Mathematical Proceedings of the Cambridge Philosophical Society. – 1956. – Vol. 52(1). – P. 17-19. – DOI 10.1017/S0305004100030929.
9. Concrete Compressive Strength / UCI Machine Learning Repository. URL: <https://archive.ics.uci.edu/dataset/165/concrete+compressive+strength> (дата обращения: 29.04.2024).
10. Harrison, Jr. Hedonic housing prices and the demand for clean air / Jr. Harrison, R. David, L. Daniel // Journal of Environmental Economics and Management. – 1978. – Vol. 5(1). – P. 81-102. – DOI 10.1017/S0305004100030929.
11. Rice (Cammeo and Osmancik) / UCI Machine Learning Repository. URL: <https://archive.ics.uci.edu/dataset/545/rice+cammeo+and+osmancik> (дата обращения: 29.04.2024).

12. Yeast / UCI Machine Learning Repository. URL: <https://archive.ics.uci.edu/dataset/110/yeast> (дата обращения: 29.04.2024).
13. Geppy: a gene expression programming framework in Python / GitHub. URL: <https://github.com/ShuhuaGao/geppy> (дата обращения: 29.04.2024).

References

1. Gimadieva, L. S. Forecasting the cost of housing / L. S. Gimadieva // Regional problems of economic transformation. – 2022. – Vol. 3(137). – P. 22-32. – DOI 10.26726/1812-7096-2022-3-22-32.
2. Bolodurina, I. P. Intelligent model for predicting vehicle traffic intensity at an intersection / I. P. Bolodurina, L. M. Antsiferova, L. S. Grishina // Intelligence. Innovation. Investments. – 2022. – Vol. 6. – P. 69-78. – DOI 10.25198/2077-7175-2022-6-69.
3. U. Ilhan. Classification of Osmancik and Cammeo Rice Varieties using Deep Neural Networks / U. Ilhan, A. Ilhan, K. Uyar, E. I. Iseri // 2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) – Ankara, Turkey, 2021. – P. 587-590. – DOI 10.1109/ISMSIT52890.2021.9604606.
4. Nikolaev, P. L. Book Genre Classification on the Base on Text Description through Deep Learning / P. L. Nikolaev // International Journal of Open Information Technologies. – 2022. – Vol. 10(1). – P. 36-40.
5. Guang-Bin Huang. Extreme learning machine: Theory and applications / Guang-Bin Huang, Qin-Yu Zhu, Chee-Kheong Siew // Neurocomputing. – 2006. – Vol. 70. – P. 489-501. – DOI 10.1016/j.neucom.2005.12.126.
6. Demidova, L.A. Biology-inspired optimization algorithms applied to intelligent input weights selection of an extreme learning machine in regression problems / L. A. Demidova, A. V. Gorchakov // AIP Conference Proceedings – 2023. – Vol. 2700(1). – DOI 10.1063/5.0124917.
7. Candida Ferreira. Gene Expression Programming: A New Adaptive Algorithm for Solving Problems / Candida Ferreira // Complex Syst. – 2001. – Vol. 13. – DOI 10.48550/arXiv.cs/0102027.
8. Penrose R. On best approximate solutions of linear matrix equations / Penrose R. // Mathematical Proceedings of the Cambridge Philosophical Society. – 1956. – Vol. 52(1). – P. 17-19. – DOI 10.1017/S0305004100030929.
9. Concrete Compressive Strength / UCI Machine Learning Repository. URL: <https://archive.ics.uci.edu/dataset/165/concrete+compressive+strength> (accessed: 29/04/2024).
10. Harrison, Jr. Hedonic housing prices and the demand for clean air / Jr. Harrison, R. David, L. Daniel // Journal of Environmental Economics and Management. – 1978. – Vol. 5(1). – P. 81-102. – DOI 10.1017/S0305004100030929.
11. Rice (Cammeo and Osmancik) / UCI Machine Learning Repository. URL: <https://archive.ics.uci.edu/dataset/545/rice+cammeo+and+osmancik> (accessed: 29/04/2024).
12. Yeast / UCI Machine Learning Repository. URL: <https://archive.ics.uci.edu/dataset/110/yeast> (accessed: 29/04/2024).
13. Geppy: a gene expression programming framework in Python / GitHub. URL: <https://github.com/ShuhuaGao/geppy> (accessed: 29/04/2024).