DSPA:

Вопросы применения цифровой обработки сигналов

№1

2025

СОДЕРЖАНИЕ

ИССЛЕДОВАНИЕ ИТЕРАЦИОННОЙ РЕАЛИЗАЦИИ АЛГОРИТМА MMSE

Бен Режеб Софиэн Бен Камель
МТУСИ, аспирант, Москва, Россия
sbenrezheb@yandex.ru

Крейнделин Виталий Борисович
МТУСИ, д.т.н., профессор, Москва, Россия
vitkrend@gmail.com

Аннотация
Исследуется итерационная реализация алгоритма MMSE с целью снижения вычислительной сложности при сохранении высокого уровня помехоустойчивости. В рамках работы рассматривается антенная конфигурация 128×128 и многолучевая среда передачи данных. Используется модуляция QAM-16, что позволяет оценить влияние итерационного подхода на вероятность ошибки на бит. Приводятся результаты анализа вычислительной сложности и помехоустойчивости, демонстрирующие возможности итерационной реализации алгоритма MMSE в условиях высоких требований к эффективности и надежности связи.

Ключевые слова
Радиосвязь, система беспроводной связи, Multiple Input Multiple Output, MMSE, OGM2, модуляция QAM-16

Введение

Современные системы беспроводной связи требуют высокой надежности и эффективности передачи данных, что особенно актуально для применения технологий с множественным входом и выходом MIMO (Multiple Input Multiple Output). Эти технологии значительно увеличивают пропускную способность каналов связи и устойчивость к многолучевым искажениям. Однако рост числа антенн в системах, например, с конфигурацией 128×128, и использование модуляции, такой как QAM-16, приводят к значительному увеличению вычислительной сложности алгоритмов обработки сигналов.

Алгоритм MMSE считается одним из ключевых методов обработки сигналов в системах MIMO благодаря своей достаточно низкой вычислительной сложности при приемлемой помехоустойчивости. Однако при большом числе антенн в системе MIMO традиционная реализация алгоритма MMSE требует значительных вычислительных ресурсов. Это становится серьезным ограничением при использовании в системах с большим количеством антенн и высокими требованиями к пропускной способности. Поэтому является актуальным исследование возможностей оптимизации алгоритма MMSE с целью снижения его вычислительной сложности [1].

Настоящее исследование посвящено изучению итерационной реализации алгоритма MMSE, которая может существенно снизить вычислительную сложность без значительной потери помехоустойчивости. Работа проводится для антенной конфигурации 128×128 и модуляции QAM-16, что позволяет оценить эффективность предложенного подхода в условиях многолучевой среды. Полученные результаты помогут определить перспективы использования итерационного алгоритма MMSE для задач обработки сигналов в современных системах беспроводной связи [2].

Постановка задачи

В данном докладе рассматривается система связи с несколькими передающими и несколькими приемными антеннами (система MIMO) в конфигурации 128×128.

Целью исследования является изучение итерационной реализации алгоритма MMSE, направленной на снижение вычислительной сложности при сохранении высокого уровня помехоустойчивости. Особое внимание уделяется анализу эффективности алгоритма в условиях многолучевой среды распространения сигнала и использования модуляции QAM-16.

В рамках работы исследуется вероятность битовой ошибки (BER) для итерационной реализации алгоритма MMSE. Выполняется оценка его вычислительной сложности и сравнение с традиционной реализацией MMSE, что позволяет определить перспективы применения итерационного подхода в системах MIMO.

Модель системы беспроводной связи

Модель системы MIMO изображена на рисунке 1. Мы представляем систему связи с числом передающих антенн M и числом приемных антенн *N*.

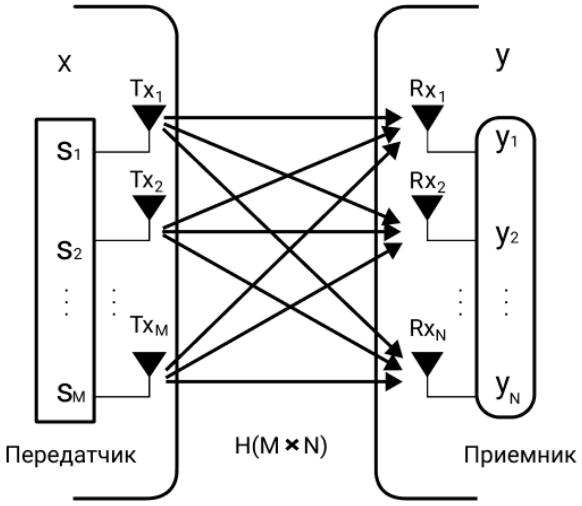


Рис. 1. Модель системы MIMO

Антенны $Tx_1,...,Tx_M$ передают сигналы $s_1,...,s_M$ на приемные антенны $Rx_1,...,Rx_N$. В каждой приемной антенне сигналы, приходящие от всех передающих антенн, складываются. Принятые сигналы в антеннах $Rx_1,...,Rx_N$ обозначаются как $y_1,...,y_N$. Выражаем принятый сигнал на антенне Rx_q,q=1,...,N, в виде:

$$Y_q = \sum_{p=1}^{M} h_{qp} \cdot x_p + n_q; q = 1,..., N. \tag{1}$$

Модель канала MIMO описывается следующим образом [3-5]:

$$\mathbf{y = Hs + n} \tag{2}$$

где **H** – это комплексная матрица канала размерности N×M , имеющая следующий вид:

$$H = \begin{pmatrix} h_{11} & h_{12} & \cdots & h_{1M} \\ h_{21} & h_{22} & & h_{2M} \\ \vdots & & \ddots & \vdots \\ h_{N1} & h_{N2} & \cdots & h_{NM} \end{pmatrix}, \tag{3}$$

где h_{qp}, $p = 1,..., M, q = 1,..., N,$ – коэффициент передачи комплексного канала, который связывает передающую антенну Tx_p с приемной антенной Rx_q [6].

- $s = [s_1,..., s_M]^T$ – комплексный вектор передаваемого сигнала размерности $M \times 1$.
- $y = [y_1,..., y_N]^T$ – комплексный вектор принятого сигнала размерности $N \times 1$.
- $n = [n_1,..., n_N]^T$ – комплексный вектор аддитивного гауссовского шума размерности $N \times 1$ [7]. Этот вектор имеет некоррелированные компоненты, каждая из которых имеет нулевое среднее и дисперсию $2\sigma_n^2$.

Алгоритм демодуляции MMSE

Метод минимизации среднеквадратической ошибки (MMSE) представляет собой один из способов обработки сигналов, основанный на линейных преобразованиях. Основная цель этого метода заключается в применении матричной операции к принятому вектору *y*, чтобы получить оценку переданных данных $\hat{\mathbf{s}}^{MMSE}$. Однако полученное значение $\hat{\mathbf{s}}^{MMSE}$ зачастую не принадлежит множеству возможных символов Θ^{I} что делает необходимым его дальнейшее преобразование. Этот этап определяется особенностями выбранного метода модуляции [8, 9].

Алгоритм MMSE вычисляет оценку $\hat{\mathbf{s}}^{MMSE}$ следующим образом:

$$\hat{\mathbf{s}}^{\mathrm{MMSE}} = \mathbf{W}_{\mathrm{MMSE}}\mathbf{y}\,, \tag{4}$$

где $\mathbf{W}_{\mathrm{MMSE}} = \left(\mathbf{H}'\mathbf{H} + 2\sigma_n^2 1\right)^{-1}\mathbf{H}'$ – матрица линейного преобразования, Параметр $2\sigma_n^2$ соответствует уровню шума, поступающего на вход каждой приемной антенны [10-12].

Предложенный оптимизированный градиентный метод 2 (OGM2)

Заменим задачу демодуляции на задачу непрерывной оптимизации некоторой скалярной функции *f*(*s*) от вектора комплексных информационных символов $\mathbf{s}$. В качестве функции *f*(*s*) выберем следующую квадратичную функцию:

$$f(\mathbf{s}) = \|\mathbf{Y} - \mathbf{H}\mathbf{s}\|^2 + 2\sigma_n^2 \cdot \|\mathbf{s}\|^2\,. \tag{5}$$

Будем искать минимум функции (4) где задача оптимизации без ограничений:

$$\min_{\mathbf{s}} f(\mathbf{s}) = \min_{\mathbf{s}}\left\{\|\mathbf{Y} - \mathbf{H}\mathbf{s}\|^2 + 2\sigma_n^2 \cdot \|\mathbf{s}\|^2\right\} = \min_{\mathbf{s}}\left\{(\mathbf{Y} - \mathbf{H}\mathbf{s})'(\mathbf{Y} - \mathbf{H}\mathbf{s}) + 2\sigma_n^2 \cdot \mathbf{s}'\mathbf{s}\right\}. \tag{6}$$

Решать задачу (4) минимизации функции $f(\mathbf{s})$ можно многими разными методами. Эти методы между собой отличаются по скорости сходимости, по сложности и т.д. В данном случае был выбран метод OGM2 (Оптимизированный градиентный метод 2 – Optimized Gradient Method 2) [13].

Алгоритм OGM2 представляет собой модификацию метода ускоренного градиентного спуска, которая обеспечивает более быструю сходимость при решении задач минимизации выпуклых функций. Этот метод относится к семейству методов первого порядка, что делает его особенно привлекательным для задач с высокой размерностью, где вычислительная сложность играет важную роль [14].

Для последующего изложения потребуются выражения вектора градиента:

$$grad\left(f(\mathbf{s})\right) = \frac{df(\mathbf{s})}{d\mathbf{s}} = 2\cdot\left(\mathbf{H}'\mathbf{H} + 2\sigma_n^2\cdot\mathbf{1}\right)\cdot\mathbf{s} - 2\cdot\mathbf{y} = 2\cdot\left(\mathbf{H}'\cdot(\mathbf{H}\cdot\mathbf{s}) + 2\sigma_n^2\cdot\mathbf{s}\right) - 2\cdot\mathbf{y} \tag{7}$$

Алгоритм OGM2 предназначен для минимизации выпуклой функции *f* с липшицевым градиентом $\left(f \in C_L^{1,1}\right)$, где *L* – константа Липшица для градиента. Исходные параметры включают начальную точку $x_0 \in \mathbb{R}^d$ и начальное значение параметра $\theta_0 = 1$ [15].

Процесс работы алгоритма включает следующие шаги:

1. Градиентное обновление:

Для каждого шага *i* вычисляется временная точка y_{i+1}:

$$y_{i+1} = x_i - \frac{1}{L}grad\left(f(x_i)\right) \tag{8}$$

$grad\left(f(x_i)\right)$ – градиент целевой функции в точке x_i.

2. Коррекция с использованием суммы градиентов: используется накопление градиентов на всех предыдущих шагах:

$$z_{i+1} = x_0 - \frac{1}{L}\sum_{k=0}^{i} 2\theta_k \, grad\left(f(x_k)\right), \tag{9}$$

где θ_k – параметр ускорения, обновляемый на каждом шаге [16].

3. Обновление параметра ускорения:

Значение θ_{i+1} задается следующими формулами:

$$\theta_{i+1} = \begin{cases} \dfrac{1+\sqrt{1+4\theta_i^2}}{2}, i \le N-2, \\ \dfrac{1+\sqrt{1+8\theta_i^2}}{2}, i = N-1. \end{cases} \tag{10}$$

4. Обновление основной точки:

Конечное значение текущего шага x_{i+1} определяется

как взвешенная комбинация точек y_{i+1} и z_{i+1} из выражений (7), (8) и (9).

$$z_{i+1} = \left(1 - \frac{1}{\theta_{i+1}}\right) y_{i+1} + \frac{1}{\theta_{i+1}} z_{i+1}. \tag{11}$$

Алгоритм OGM2, описываемый выражениями (7)…(10), достигает ускоренной сходимости благодаря использованию дополнительной корректировки через z_{i+1}, которая учитывает накопленные градиенты. Эта модификация удваивает вес всех предыдущих градиентов, что повышает эффективность метода.

Скорость сходимости алгоритма OGM2 соответствует лучшим теоретическим пределам для методов первого порядка. Несмотря на добавление новой компоненты z_{i+1}, алгоритм требует сопоставимого объема вычислений с аналогами, такими как метод Nesterov или FGM2 (Быстрый градиентный метод – Fast Gradient Method 2) [17-19].

Вычислительная сложность

Эффективность алгоритма в значительной степени определяется его вычислительной сложностью, которая включает количество основных операций: умножений, сложений и делений. В системах с высокой размерностью, таких как MIMO с конфигурацией $M \times N$, эти затраты существенно влияют на скорость обработки данных и потребление ресурсов.

Для оценки вычислительной сложности алгоритмов OGM2 (7)…(10) и MMSE (3) выполнен подробный анализ, включающий подсчет количества операций для каждого алгоритма при заданных условиях. Учитывалось количество итераций I, необходимое для достижения вероятности ошибки на бит 5×10^2, а также размер антенной решетки $M = N$.

Результаты расчетов представлены в Таблице 1, где указано количество операций каждого типа (умножения, сложения и деления), а также общее число операций для обоих алгоритмов [20].

Таблица 1

Вычислительная сложность OGM2 и MMSE

Итераций	16	32	-
	Алгоритм OGM2	Алгоритм OGM2	MMSE
Умножений	$I(8M^2+16M+10)+4M^2+2M$ 2,195,872	$I(8M^2+16M+10)+4M^2+2M$ 4,323,952	$4M^3+6M^2-M$ 8,486,784
сложений	$I(8M^2+8M+5)+2M(2M-1)$ 2,176,896	$I(8M^2+8M+5)+2M(2M-1)$ 4,291,712	$4M^3+3M^2-3M$ 8,437,376
делений	-	-	$M+1$ 129
Итого	4,372,768	8,615,664	16,924,289

Результаты моделирования

Статистическое моделирование было проведено для исследования помехоустойчивости системы MIMO. Условия моделирования:

- Конфигурация антенн – 128 передающих и 128 приемных антенн.
- Число итераций в алгоритме OGM2 (7)…(10) – 32 и 16.
- Число экспериментов – 1000.
- Независимые рэлеевские замирания.
- Модуляция – QAM-16.
- Алгоритмы демодуляции – MMSE и OGM2.

Рассмотрим характеристики системы MIMO с реализованными алгоритмами MMSE и OGM2. Сравним эти характеристики с точки зрения отношения числа ошибочно принятых битов к общему числу принятых битов (Bit Error Rate, BER) для разных значений отношения сигнал/шум (Signal to Noise Ratio, SNR).

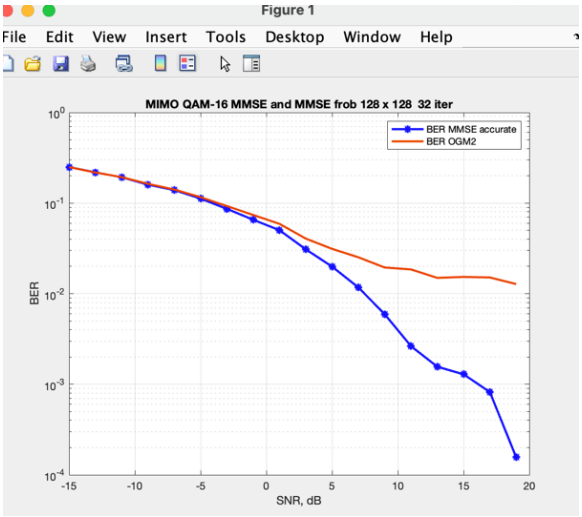


Рис. 2. Характеристики системы MIMO при 16 итераций и антенной конфигурации 128x128

На рисунке 2 представлены результаты моделирования вероятности битовой ошибки (BER) для системы MIMO с использованием алгоритмов MMSE и OGM2 при 32 итерациях. Из графика видно, что алгоритм MMSE демонстрирует более низкий уровень BER по сравнению с OGM2 для всех значений отношения сигнал/шум (SNR). Энергетические потери алгоритма OGM2 заметны уже при BER порядка 10^{-2}, и с увеличением SNR разница между алгоритмами становится более существенной.

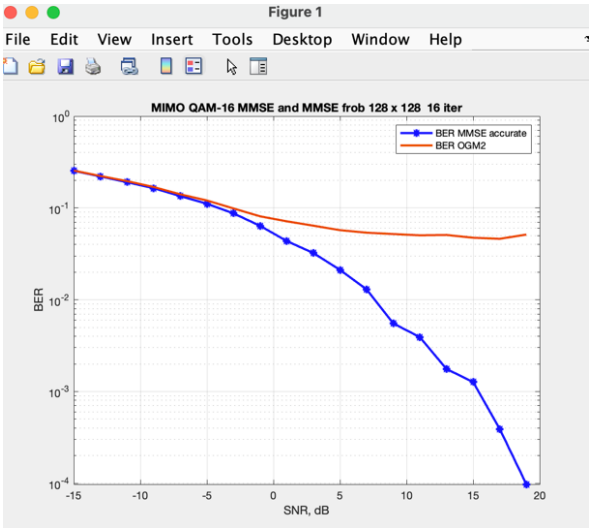


Рис. 3. Характеристики системы MIMO при 32 итераций и антенной конфигурации 128x128

На рисунке 3 представлены результаты моделирования для тех же алгоритмов при сокращении числа итераций до 16. Здесь наблюдается снижение помехоустойчивости алгоритма OGM2, особенно в области высоких значений SNR.

Из рисунка 2 видно, что потери в помехоустойчивости у алгоритма OGM2 при числе итераций 32 по сравнению с алгоритмом MMSE появляются при $BER < 5 \cdot 10^{-2}$. Если $BER > 5 \cdot 10^{-2}$, то потери в помехоустойчивости оказываются незначительными. При этом из Таблицы 1 следует, что алгоритм OGM2 имеет выигрыш в вычислительной сложности в 2 раза по сравнению с алгоритмом MMSE.

Из рисунка 3 видно, что потери в помехоустойчивости у алгоритма OGM2 при числе итераций 16 по сравнению с алгоритмом MMSE появляются уже при $BER < 10^{-1}$. Если $BER > 10^{-1}$, то потери в помехоустойчивости также оказываются незначительными. При этом из таблицы 1 следует, что алгоритм OGM2 имеет выигрыш в вычислительной сложности в четыре раза по сравнению с алгоритмом MMSE.

Заключение

Результаты моделирования подтвердили, что использование итерационного подхода позволяет значительно снизить вычислительную сложность, что особенно важно для систем MIMO с большим числом антенн, таких как MIMO 128×128 с модуляцией QAM-16.

Тем не менее, сокращение числа итераций приводит к ухудшению помехоустойчивости, особенно при высоких значениях SNR. В то же время, путем выбора числа итераций в алгоритме OGM2 можно найти приемлемый компромисс между вычислительной сложностью и помехоустойчивостью алгоритма демодуляции.

В заключение можно отметить следующие ключевые моменты:

- Итерационный подход в алгоритме OGM2 позволяет снизить вычислительную сложность, что критично для систем передачи данных с большим числом антенн.
- Итерационный алгоритм OGM2, несмотря на меньшую сложность, уступает алгоритму MMSE по помехоустойчивости. Однако путем выбора оптимального числа итераций можно достичь приемлемого компромисса между помехоустойчивостью и вычислительной сложностью.
- В качестве направления дальнейших исследований можно указать на необходимость выбора итерационных алгоритмов решения оптимизационной задачи (5), обладающих более высокой скоростью сходимости по сравнению с алгоритмом OGM2. Это позволило бы достичь еще более существенного снижения вычислительной сложности алгоритма демодуляции.

Литература

1. *Bara'u Gafai Najashi and Tan Xiaoheng*. A Comparative Performance Analysis of Multiple-Input Multiple-Output using MATLAB with Zero Forcing and Minimum Mean Square Error Equalizers, American J. of Engineering and Applied Sciences 4 (3), pp. 425-428, 2011.

2. *Shaoshi Y., Lajos H*. Fifty Years of MIMO Detection: The Road to Large-Scale MIMOs. Accepted to appear on IEEE communications surveys & tutorials, 2015, 24.

3. *Huang H., Papadias C.B., Venkatesan S*. MIMO Communications for Cellular Networks. USA, Springer Science Busyness Media LLC, 2012. 314 p.

4. *Бакулин М.Г., Варукина Л.А., Крейнделин В.Б.* Технология MIMO. Принципы и алгоритмы. М.: Горячая линия – Телеком, 2014. 244 с.

5. *Sofien B.R., Kreyndelin V*. Investigation of the Noise Immunity of MMSE and ZF Algorithms in MIMO Systems under Conditions of Correlated Fading // 2023 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF), St. Petersburg, Russian Federation, 2023, pp. 1-5, doi: 10.1109/WECONF57201.2023.10147916.

6. *Sofien B.R., Kreyndelin V*. Study of Interference Immunity of MMSE, ZF, and ML Demodulation Algorithms in MIMO Systems Under High-Order Modulation // 2023 Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO, Pskov, Russian Federation, 2023, pp. 1-5, doi: 10.1109/SYNCHROINFO57872.2023.10178581.

7. *Крейнделин В.Б., Смирнов А.Э., Бен Режеб Т.Б.К.* Исследование радиоинтерфейса беспроводных систем межмашинного взаимодействия M2M // T-Comm: Телекоммуникации и транспорт. 2014. Т. 8, № 6. С. 71-74. EDN SGWEHR

8. *Шлома А.М., Бакулин М.Г., Крейнделин В.Б., Шумов А.П*. Новые технологии в системах мобильной радиосвязи М.: Московский технический университет связи и информатики, 2005. 455 с. EDN YQQJQN

9. *Nechaev Y.B., Dvorakova I.O., Malyutin A.A*. Fast channel estimation algorithm for HF MIMO system // 2014 24th International Crimean Conference Microwave & Telecommunication Technology, Sevastopol, Ukraine, 2014, pp. 352-353, doi: 10.1109/CRMICO.2014.6959428

10. *Bacci G., Alberto D'Amico A., Sanguinetti L*. MMSE Channel Estimation in Large-Scale MIMO: Improved Robustness with Reduced Complexity // IEEE Transactions on Wireless Communications, vol. 23, no. 12, pp. 18563-18575, Dec. 2024, doi: 10.1109/TWC.2024.3470124.

11. *Kwan M.-W., Kok C.-W*. MMSE Equalizer for MIMO-ISI Channel with Shorten Guard Period // IEEE Transactions on Signal Processing, vol. 55, no. 1, pp. 389-395, Jan. 2007, doi: 10.1109/TSP.2006.882067.

12. *Osinsky A., Ivanov A., Lakontsev D., Bychkov R., Yarotsky D*. Data-Aided LS Channel Estimation in Massive MIMO Turbo-Receiver // 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 2020, pp. 1-5, doi: 10.1109/VTC2020-Spring48590.2020.9128566.

13. *Bacci G., Alberto D'Amico A., Sanguinetti L*. MMSE Channel Estimation in Large-Scale MIMO: Improved Robustness with Reduced Complexity // IEEE Transactions on Wireless Communications, vol. 23, no. 12, pp. 18563-18575, Dec. 2024, doi: 10.1109/TWC.2024.3470124.

14. *Yoldas Y., Goren S., Onen A*. Optimal Control of Microgrids with Multi-stage Mixed-integer Nonlinear Programming Guided Q-learning Algorithm // Journal of Modern Power Systems and Clean Energy, vol. 8, no. 6, pp. 1151-1159, November 2020, doi: 10.35833/MPCE.2020.000506.

15. *Goh J., Kim K.* High Efficiency 1700V 4H-SiC UMOSFET with Local Floating Superjunction // 2020 International Conference on Electronics, Information, and Communication (ICEIC), Barcelona, Spain, 2020, pp. 1-5, doi: 10.1109/ICEIC49074.2020.9051234.

16. *Adesuyi T.A., Kim B.M.* Preserving Privacy in Convolutional Neural Network: An ∈-tuple Differential Privacy Approach // 2019 IEEE 2nd International Conference on Knowledge Innovation and Invention (ICKII), Seoul, Korea (South), 2019, pp. 570-573, doi: 10.1109/ICKII46306.2019.9042653.

17. *Rusen V., Krukonis A., Plonis D.* Prediction of Parameters of Semiconductor Band-pass Filters using Artificial Neural Network // 2020 IEEE 8th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE), Vilnius, Lithuania, 2021, pp. 1-4, doi: 10.1109/AIEEE51419.2021.9435748.

18. *Nesterov Y*. Lectures on Convex Optimization. Second Edition, CORE/INMA Catholic University of Louvain Louvain-la-Neuve, Belgium, 2018.

19. *Жадан В.Г*. Методы оптимизации. Часть II. Численные алгоритмы," учебное пособие, М.: МФТИ, 2015, 320 с.

20. *Бакулин М.Г., Крейнделин В.Б., Панкратов Д.Ю*. Технологии в системах радиосвязи на пути к 5G. М.: Горячая линия – Телеком", 2018. 280 с.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ КЛЮЧЕВЫХ ОСОБЕННОСТЕЙ ГЕНЕРАТИВНО-СОСТЯЗАТЕЛЬНЫХ СЕТЕЙ

Фатхулин Тимур Джалилевич,
Московский технический университет связи и информатики, доцент кафедры МК и ИТ, к.т.н., Москва, Россия
t.d.fatkhulin@mtuci.ru

Боданюк Алексей Павлович,
Московский технический университет связи и информатики, студент группы БВТ2302, Москва, Россия

Рахматова Азиза Акрамовна,
Московский технический университет связи и информатики, студентка группы М62402, Москва, Россия

Аннотация
В работе рассмотрены генеративно-состязательные сети (GAN), представляющие собой классы нейронных сетей, способные создавать реалистичные синтетические данные. Цель работы – определить преимущества и недостатки генеративно-состязательных сетей. Проанализированы различные архитектуры GAN, включая стандартные GAN, Deep Convolutional GAN (DCGAN), Conditional GAN (cGAN) и их модификации, а также их ключевые особенности. В заключении сделаны выводы о применимости той или иной архитектуры для практических задач. Методологической основой статьи являются описательный метод, методы теоретического анализа, а также метод обобщения.

Ключевые слова
Генеративно-состязательные сети, архитектура, синтетические данные, изменение цветности, устойчивость, переобучение, сравнительный анализ

Введение

Генеративно-состязательные сети (Generative Adversarial Networks, GAN) были предложены Яном Гудфеллоу в 2014 году [1, 17-19]. Они представляют собой важное достижение в машинном обучении (МО) [22-26, 28-30]. Идея модели состоит в том, что используются две соревнующиеся нейронные сети – генератор и дискриминатор, которые обучаются в процессе, напоминающем игру с нулевой суммой. Генератор создает новые данные, стремясь приблизить их к распределению оригинального набора, тогда как дискриминатор пытается отличить сгенерированные данные от реальных. Процесс обучения GAN направлен на улучшение распознавания подделок дискриминатором и создание более реалистичных данных генератором [2, 27].

GAN получили широкое применение благодаря способности генерировать изображения, видео и тексты высокого качества, что позволило использовать их для задач автоматического раскрашивания, синтеза текстур и восстановления изображений [3]. Тем не менее, эти модели также сталкиваются с трудностями, такими как сложность обучения и склонность к переобучению.

В данной работе проведен сравнительный анализ базовых Generative Adversarial Networks, Deep Convolutional Generative Adversarial Network (глубокая сверточная генеративная состязательная сеть), Conditional Generative Adversarial Network (условная генеративная состязательная сеть), Cycle Generative Adversarial Network (циклическая генеративная состязательная сеть) и Pix2Pix Generative Adversarial Network (от пикселя к пикселю генеративная состязательная сеть) [20-25, 28-30] для выявления их особенностей, преимуществ и ограничений.

Анализ составляющих архитектуры GAN

Архитектуры генератора и дискриминатора построены по принципу многослойного перцептрона [4, 26]. Поскольку такие задачи, как колоризация, относятся к классу преобразования изображений, генератор и дискриминатор являются сверточными нейронными сетями (Convolutional Neural Net-

works). Генератор (G) представлен отображением $G(z;\theta_G)$, где z – равномерно распределенная переменная шума, которая используется в качестве входных данных для генератора. Аналогично, дискриминатор (D) представлен отображением $D(x;\theta_D)$, которое выдает скаляр в диапазоне от 0 до 1, где x – цветное изображение. Выход дискриминатора интерпретируется как вероятность того, что входные данные принадлежат обучающему набору. Это позволяет определить задачу оптимизации: G обучается минимизировать вероятность того, что дискриминатор правильно классифицирует данные, а D - максимизировать вероятность правильной классификации. Математически это выражается следующим образом:

$$\min_{\theta_G} J^{(G)}(\theta_D,\theta_G) = \min_{\theta_G} E_z[\log(1 - D(G(z))], \max_{\theta_D} J^{(D)}(\theta_D,\theta_G) = \max_{\theta_D}(E_z[\log(D(x))] + E_z[\log(1 - D(G(z)))]).$$

Эти два уравнения задают функции стоимости, необходимые для обучения GAN. В литературе их часто объединяют в одну задачу минимаксной игры с функцией стоимости $V(G,D)$:

$$\min_G \max_D V(G,D) = E_x[\log D(x)] + E_z[\log(1 - D(G(z)))].$$

Однако этот подход имеет два недостатка. Если дискриминатор хорошо себя показывает в процессе обучения, то градиент генератора во время обратного распространения будет близок к нулю, что значительно уменьшит скорость сходимости, так как генератор будет продолжать выдавать очень схожие результаты. А также предложенная функция стоимости не ограничена снизу, что может привести к расхождению к « $-\infty$ » в процессе минимизации.

Архитектуры Conditional GAN и DCGAN

В классической архитектуре GAN входными данными генератора является случайный шум z. Однако этот подход не применим к таким задачам, как, например, колоризация изображений. Эта проблема была решена за счет *Conditional GAN (CGAN)* [5]. Поскольку шум не вводится, вход генератора рассматривается как нулевой шум с градациями серого в качестве априорной информации (условие y). В таком случае задача оптимизации будет выглядеть так:

$$\min_G \max_D V(G,D) = E_x[\log D(x \mid y)] + E_z[\log(1 - D(G(z \mid y)))].$$

Дискриминатор получает цветные изображения как от генератора, так и из исходных данных, вместе с градациями серого в качестве условия, и пытается определить, какая пара содержит истинное цветное изображение. Колоризация изображений представляет собой задачу перевода изображения в изображение (image-to-image translation), в которой отображение происходит из пространства высокоразмерного ввода в высокоразмерный вывод. Это можно рассматривать как задачу поканальной регрессии, где структура входных данных тесно согласована со структурой выходных. Это означает, что сеть должна не только генерировать вывод с такими же пространственными размерами, как вход, но и добавлять цветовую информацию к каждому пикселю входного изображения в градациях серого.

Несмотря на применимость Conditional GAN в задачах колоризации изображений, у этой архитектуры есть несколько недостатков. Первый – устойчивость к обучению: как и классическая архитектура, CGAN склонна к нестабильности обучения из-за сложности минимаксной задачи, особенно при работе с высокоразмерными [6] данными. Второй – зависимость от условия: если черно-белое изображение y содержит недостаточно информации о текстуре и структуре, то генератор может добавлять некорректные цвета.

Для решения вышеуказанных проблем была предложена архитектура *Deep Convolutional GAN (DCGAN)* [7]. Использование сверточных сетей позволяет лучше учитывать пространственную структуру и сохранять информацию между слоями. У этой архитектуры есть несколько принципиальных отличий от классических GAN.

Во-первых, заменены пулинговые (*термин от англ. – pooling*) слои: вместо базовых операций пулинга (max-pooling, average-pooling) для уменьшения размерности изображения используются сверт-

ки с отступом (strided convolutions) в дискриминаторе и частично-страйдинговые свертки (fractional-strided convolutions) в генераторе. Это изменения способствует лучшему качеству понижения и повышения размерности изображений [8].

Во-вторых, нормализация по батчам (термин от англ. - batch): она применяется во всех слоях моделей, кроме входного слоя дискриминатора и входного для генератора. Нормализация батчей делает обучение более стабильным и улучшает сходимость в процессе обучения [9].

В-третьих, отказ от полносвязных слоев и изменение функций активации: в DCGAN были удалены полносвязные слои, что позволило использовать более глубокие архитектуры без усложнения. А также в генераторе для всех слоев, кроме последнего, используется функция активации ReLU, а для дискриминатора LeakyReLU.

Вышеуказанные изменения помогли добиться лучшей генерации изображений, с более естественными текстурами и цветами [10]. Благодаря нормализации батчей и замене пулинга удалось повысить устойчивость обучения, а также использование сверточных слоев позволило более эффективно обрабатывать высокоразмерные изображения. Однако ConditionalDCGAN (Conditional Deep Convolutional GAN) все ещё сильно зависим от входного условия и требователен к ресурсам. Также для успешной работы будет требоваться тонкая настройка гиперпараметров [11].

Архитектура CycleGAN: новый подход к image-to-image translation

Для решения задач без использования парных данных была предложена архитектура *CycleGAN* [12]. Например, для колоризации изображений можно обучить модель преобразовывать картинки из домена черно-белых в домен цветных. Архитектура CycleGAN состоит из двух генераторов и двух дискриминаторов. Генератор G преобразует изображения из домена X в домен Y, а генератор F – наоборот. Ключевой особенностью является функция циклической согласованности (cycle-consistency loss), которая занимается минимизацией разницы между исходным изображением и восстановленным после двойного преобразования.

$$\mathrm{L}_{\text{cyc}}(G,F) = E_{x\sim p_{data}(x)}[\| F(G(x)) - x \|\|] + E_{y\sim p_{data}(y)}[\| G(F(y)) - y \|\|]$$

В этой архитектуре задача оптимизации выглядит так:

$$G^*, F^* = \arg\min_{G,F}\max_{\mathrm{D_X},D_Y} \mathrm{L}_{\text{GAN}}(G,F,D_X,D_Y)$$

Эта архитектура не зависит от парных данных, что делает её очень гибкой, модель может быть использована для колоризации, стилизации, преобразования погоды на изображениях и множества других задач [13]. Она позволяет добиваться высокого уровня детализации и реализма, а также предоставляет возможность управления стилем, что особенно необходимо для задач с варьированием цветовых схем.

Эта архитектура очень требовательна к ресурсам [14], так как она использует двойной набор генераторов и дискриминаторов. Также из-за отсутствия начального условия, как в ConditionalDCGAN или ConditionalGAN, точность колоризации может быть хуже. Тем не менее, эта архитектура является универсальной во множестве задач image-to-image translation.

Pix2Pix [15] – условный GAN для image-to-image преобразований. Эта архитектура была предложена одной из первых для задач перевода изображений. В отличие от CycleGAN, в Pix2Pix используются парные наборы данных, и там минимизируется разница между целевыми и предсказанными изображениями.

Основа архитектуры Pix2Pix построена на ConditionalGAN, который был разобран ранее. Однако в этой вариации структура генератора представляет из себя U-Net [13], а дискриминатора – PatchGAN, который оценивает качество изображения не целиком, а по нескольким отдельным патчам. Задача оптимизации включает в себя 2 нижеприведенные составляющие.

Adversarial loss:

$$L_{GAN}(G,D) = E_{x,y}[\log D(x,y)] + E_x[\log(1 - D(x,G(x)))]$$ *Reconstruction loss:*

$$L_{L1}(G) = E_{x,y}[\| y - G(x) \|]$$

Эта функция минимизирует разницу между истинным и предсказанным изображениями.

Благодаря наличию парных наборов данных, Pix2Pix может генерировать изображения высокой точности, а также эта архитектура подходит для широкого спектра задач. Несмотря на это, создание парных наборов данных занимает намного больше времени, что является определенным минусом. Также невзирая на использование PatchGAN, модель может генерировать повторяющиеся паттерны, это проблема особенно заметно на изображениях с большим разрешением.

В целом можно выделить эту архитектуру как мощный инструмент для работы с image-to-image преобразованиями, у неё есть как преимущества, так и недостатки, но это не препятствует её использованию.

Далее в таблице 1 представлен итоговый сравнительный анализ рассмотренных архитектур: Conditional GAN, Deep Convolutional GAN, Cycle GAN, Pix2Pix GAN. Сравнение произведено по следующему ряду критериев:

1. качество генерации: визуальная реалистичность, точность передачи цветов и текстур;
2. гибкость: фактор применимости к различным задачам;
3. устойчивость обучения: способность к стабильному схождению;
4. вычислительные ресурсы: требование к памяти и вычислительным мощностям;
5. зависимость от парных наборов данных;
6. сложность настройки: требования и чувствительность к гиперпараметрам.

Таблица 1

Сравнение особенностей архитектур

Архитектура / **Критерии**	***CGAN***	***DCGAN***	***CycleGAN***	***Pix2Pix***
Качество генерации	среднее	среднее	средняя	высокая
Гибкость	узкая	широкая	очень широкая	узкая
Устойчивость обучения	низкая	средняя	средняя	высокая
Ресурсоемкость	низкая	средняя	высокая	средняя
Зависимость от парных наборов	есть	нет	нет	есть
Сложность настройки	простая	средняя	сложная	средняя

Также в таблице 2 приведена применимость данных архитектур в задачах колоризации изображений.

Таблица 2

Применимость для задач колоризации

Архитектура	**Реализм цветов**	**Учет текстур**	**Стабильность**
CGAN	средний	средний	средняя
DCGAN	средний	высокий	средняя
CycleGAN	высокий	высокий	средняя
Pin2Pix	высокий	высокий	высокая

Заключение

Таким образом, в результате анализа каждой из архитектур были определены их функциональные возможности, а также преимущества и недостатки. Полученные результаты были систематизированы и представлены в виде сравнительных таблиц.

CGAN изначально была предложена для выполнения задач с задающимися условиями. Хотя она и эффективно может себя проявить для задач колоризации изображений, но устойчивость обучения и качество выходных данных не являются желаемыми. Также зависимость от парных данных определенно не является преимуществом данной архитектуры.

DCGAN ощутимо улучшает качество генерации за счет использования сверточных слоев, а нормализация обеспечивает более стабильное обучение. Несмотря на более эффективную работу с высокоразмерными изображениями, DCGAN все ещё крайне чувствителен к выбору гиперпараметров, и может испытывать трудности в задачах перевода изображений.

CycleGAN благодаря функции циклической согласованности может работать без наборов парных данных. Это особенно хорошо в тех задачах, где невозможно собрать аннотированные пары. Однако из-за использования двух наборов дискриминаторов и генераторов, у этой архитектуры очень большая ресурсоемкость.

Pix2Pix является эффективной архитектурой для задач с парными наборами данных. Использование U-Net и PatchGAN обеспечивает высокий уровень точности и детализации, однако на высокоразмерных изображениях все ещё могут возникать повторяющиеся паттерны.

Лучший выбор для колоризации – Pix2Pix. Так как в этой задаче довольно просто получить парные наборы данных, предложенная архитектура будет иметь наибольшее качество.

Однако, если речь идет об отсутствии парных наборов данных, то самое лучшее качество будет выдавать CycleGAN, а наиболее оптимальной архитектурой в отношении качество/ресурсоемкость будет DCGAN.

Литература

1. *Goodfellow I.* Generative Adversarial Nets // Advances in Neural Information Processing Systems, 2014, pp. 2672-2680.

2. *Zhang R.* Colorful Image Colorization // European Conference on Computer Vision, 2016, pp. 649-666.

3. *Karras T.* Progressive Growing of GANs for Improved Quality, Stability, and Variation [Электронный ресурс] // arXiv preprint arXiv:1710.10196, 2017. Режим доступа: https://arxiv.org/abs/1710.10196 (дата обращения: 17.11.2024).

4. *Ruck D.W., Rogers S.K., Kabrisky M., Oxley M.E., Suter B.W.* The Multilayer Perceptron as an Approximation to a Bayes Optimal Discriminant Function // IEEE Transactions on Neural Networks, 1990. Vol. 1. No. 4, pp. 296-298. DOI: 10.1109/72.80266.

5. *Mirza M., Osindero S.* Conditional Generative Adversarial Nets [Электронный ресурс] // arXiv preprint arXiv:1411.1784, 2014. Режим доступа: https://arxiv.org/abs/1411.1784 (дата обращения: 17.11.2024).

6. *Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y.* Generative Adversarial Networks // Advances in Neural Information Processing Systems, 2014.

7. *Radford A., Metz L., Chintala S.* Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks [Электронный ресурс] // arXiv preprint arXiv:1511.06434, 2016. Режим доступа: https://arxiv.org/abs/1511.06434 (дата обращения: 17.11.2024).

8. *Dumoulin V., Visin F.* A Guide to Convolution Arithmetic for Deep Learning [Электронный ресурс] // arXiv preprint arXiv:1603.07285, 2016. Режим доступа: https://arxiv.org/abs/1603.07285 (дата обращения: 17.11.2024).

9. *Ioffe S., Szegedy C.* Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift // Proceedings of the 32nd International Conference on Machine Learning, 2015.

10. *Radford A., Metz L., Chintala S.* Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks // arXiv preprint arXiv:1511.06434, 2016. Режим доступа: https://arxiv.org/abs/1511.06434 (дата обращения: 17.04.2021).

11. *Lucic M., Kurach K., Michalski M., Gelly S., Bousquet O.* Are GANs Created Equal? A Large-Scale Study // Advances in Neural Information Processing Systems (NeurIPS), 2018, pp. 700-709. DOI: 10.48550/arXiv.1711.10337

12. *Zhu J.-Y., Park T., Isola P., Efros A.A.* Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks // Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2223-2232. DOI: 10.1109/ICCV.2017.244.

13. *Karras T., Laine S., Aila T.* Analyzing and Improving the Image Quality of StyleGAN [Электронный ресурс] // arXiv preprint arXiv:1912.04958, 2019. Режим доступа: https://arxiv.org/abs/1912.04958 (дата обращения: 17.11.2024).

14. *Karras T., Aila T., Laine S., Lehtinen J.* A Style-Based Generator Architecture for Generative Adversarial Networks // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4401-4410. DOI: 10.1109/CVPR.2019.00453.

15. *Isola P., Zhu J.-Y., Zhou T., Efros A.A.* Image-to-Image Translation with Conditional Adversarial Networks // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1125-1134. DOI: 10.1109/CVPR.2017.632.

16. *Ronneberger O., Fischer P., Brox T.* U-Net: Convolutional Networks for Biomedical Image Segmentation [Электронный ресурс] // arXiv preprint arXiv:1505.04597, 2015. Режим доступа: https://arxiv.org/abs/1505.04597 (дата обращения: 17.11.2024).

17. *Вострикова П.В., Рыбка С.О., Рыжкова У.С., Фатхулин Т.Д.* Анализ нейросетевых технологий, использующихся для улучшения качества изображений // REDS: Телекоммуникационные устройства и системы. – 2024. Т. 14, № 1. С. 57-65. EDN WVBDRN

18. *Фатхулин Т.Д., Смирнов Д.А., Разумов И.В.* и др. Анализ влияния составляемых текстовых запросов (промптов) на качество изображений, генерируемых нейросетевыми технологиями // Системы синхронизации,

формирования и обработки сигналов. 2024. Т. 15, № 2. С. 52-57. EDN TSVMSK

19. *Леохин Ю.Л., Фатхулин Т.Д.* Разработка методов и алгоритма формализации текстового запроса к онлайн-сервисам, генерирующим изображения посредством нейросетевых технологий // Вестник Рязанского государственного радиотехнического университета. 2023. № 85. С. 82-95. DOI 10.21667/1995-4565-2023-85-82-95. EDN PZWYZV

20. *Леохин Ю.Л., Фатхулин Т.Д., Кожанов М.С.* Анализ и исследование применения нейросетевых технологий для генерации программного кода // Вестник Рязанского государственного радиотехнического университета. 2024. № 87. С. 41-53. DOI 10.21667/1995-4565-2024-87-41-53. EDN HKEOFX

21. *Леохин Ю.Л., Фатхулин Т.Д., Ментус М.В.* Разработка и применение методов распознавания зашумленных аудиофайлов посредством нейросетевых технологий // Вестник Рязанского государственного радиотехнического университета. 2024. № 88. С. 65-73. DOI 10.21667/1995-4565-2024-88-65-73. EDN NMXASI

22. *Маслов К.В., Фатхулин Т.Д., Иванов Д.А.* Анализ технологий автоматизации бизнес-процессов и разработки программного обеспечения с использованием low-code платформ // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. – 2024. – № 1. – С. 6-11. – EDN HDBOYM

23. *Фатхулин Т.Д., Бойцов К.В.* Анализ функционала программного обеспечения, применяемого для классификации труб на предприятии методами компьютерного зрения // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2024. № 2. С. 6-12. EDN FVQYQA.

24. *Мяличева А.А., Фатхулин Т.Д.* Анализ методов машинного обучения для прогнозирования дефектов в исходном коде // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2024. № 2. С. 16-19. EDN IVJCZF

25. *Фатхулин Т.Д., Леонова В.О., Тремасова Л.А.* Анализ нейросетевых технологий, применяемых для web-разработки // REDS: Телекоммуникационные устройства и системы. 2024. Т. 14, № 2. С. 35-41. EDN SDCNKM

26. *Фатхулин Т.Д., Исаев А.В.* Анализ моделей ARIMA и LSTM, используемых для прогнозирования криптовалют и определения портфеля инвестиций // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2024. № 2. С. 20-25. EDN ODWOPA

27. *Фатхулин Т.Д., Фатхулина Г.Г., Ментус М.В.* Разработка методики формирования запроса к нейросети с целью генерации изображений с учетом рекомендаций компьютерной лингвистики // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2023. № 1. С. 133-139. EDN PPRTOM

28. *Фатхулин Т.Д., Исаев А.В.* Анализ эффективности использования моделей ARIMA для прогнозирования котировок и определения портфеля инвестиций в области криптовалюты // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2024. № 2. С. 26-31. EDN CESRTK

29. *Фатхулин Т.Д., Бойцов К.В.* Оценка эффективности алгоритма на основе YOLO v.8 для классификации труб на предприятии по фото в зависимости от различных условий // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2024. № 2. С. 51-55. EDN KWLMYA

30. *Вишневский В.М., Леохин Ю.Л., Фатхулин Т.Д., Занегин А.В.* Методы машинного обучения в решении задачи прогнозирования спроса на отдельные виды товаров // T-Comm: Телекоммуникации и транспорт. 2024. Том 18. №10. С. 34-43.

ПРИМЕНЕНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ В БАНКОВСКОМ СКОРИНГЕ

Ванина Маргарита Федоровна
Московский технический университет связи и информатики, доцент, к.т.н., доцент, Москва, Россия
margo.vanina2012@yandex.ru

Ерохин Андрей Густавович
Московский технический университет связи и информатики, доцент, к.т.н., доцент, Москва, Россия
andrew145@yandex.ru

Аннотация
Эффективность маркетингового процесса напрямую влияет на успешность бизнеса любой компании. Одним из инновационных инструментов маркетинга в банковской сфере является использование скоринга. Банковский скоринг – это система оценки клиентов, основанная на использовании статистических методов и ставящая своей целью решение задачи о возможности выдачи банковского кредита этим клиентам. При проведении маркетинговых исследований широкое применение находят такие инновационные инструменты, как обработка больших данных, использование искусственного интеллекта, машинное обучение. В настоящей работе рассматриваются некоторые теоретические вопросы и практические примеры применения алгоритмов машинного обучения для повышения эффективности бизнес-процессов работы кредитных аналитиков в банках.

Ключевые слова
Скоринг, машинное обучение, искусственный интеллект, бизнес-процессы, кредитная аналитика, маркетинг, инновации, статистика, скоринг

Введение

Успешность любой компании в современном мире во многом зависит от эффективности организации маркетингового процесса [2]. При проведении маркетинговых исследований широкое применение находят такие инструменты, как Data Scientist, Machine Learning [3, 4] и др.

Одним из инновационных инструментов маркетинга является скоринг [1].

Чаще всего в банковской скоринг используется в банковской сфере. Банковский скоринг – это система оценки клиентов, основанная на использовании статистических методов и ставящая своей целью решение задачи о возможности выдачи банковского кредита этим клиентам [5]. Но этим применение скоринговых моделей в бизнесе не ограничивается. В более широком понимании скоринг можно охарактирозвать, как модель классификации клиентской базы на различные группы [2]. Поскольку в основе скоринговых систем лежит предположение, что люди со схожими социальными показателями ведут себя одинаково, возможно построение таких статистических моделей, которые будут полезными при ведении любого бизнеса. При этом главной проблемой при построении скоринговой модели является определение списка параметров, включаемых в скоринговую модель и определение их весов. В последнее время решение этих задач стало проще за счет привлечения технологий искусственного интеллекта.

Виды скоринговых моделей

Современные скоринговые системы участвуют во всех этапах оценки кредитоспособности: от проверки заявки на получение кредита до его полного погашения. В настоящее время наибольшее распространение получили следующие виды моделей:

- анкетный скоринг;
- противомошеннический скоринг;
- коллекторский скоринг;
- поведенческий скоринг.

Анкетный скоринг (Application scoring) – является наиболее широко используемым видом скоринга. В основе анкетного скоринга лежит процесс обработки информации о клиенте, собранной посредством анкетирования этого клиента. Результатом является вывод конечного результата об одобрении или отказе в займе потенциальному заёмщику. Методика анкетного скоринга подробно рассмотрена,

в частности, в работе [6].

Противомошеннический скоринг (Fraud scoring) представляет собой вид скоринга, предназначенный для выявления мошеннических действий. В основу этого вида скоринга положена система расчета баллов мошенничества в зависимости от базовых правил, определяющих активность пользователей, как низкий, средний, высокий или очень высокий риск. Это позволяет вычислить интегрированный показатель мошенничества потенциального клиента, и на основе этого принять решение о возможности дальнейшего сотрудничества. Технологии Fraud scoring подробно рассмотрены, в частности, в работе [7]. Данный вид скоринга часто используется в комбинации с Application scoring.

Коллекторский скоринг, или скоринг взыскания (Collection scoring) применяется при работе с клиентами, имеющими задолженность по платежам. В основу данного вида скоринга положена методика оценки возможности получения денежных средств с заемщика, на основе которой рассчитывается вероятность благоприятного исхода и формируется последовательность действий коллекторского отдела. Вопросы использования коллекторского скоринга рассматриваются, например, в работе [8]. Считается, что благодаря этому скорингу клиенты возвращают до 40% задолженностей. Также система коллекторского скоринга позволяет оценить эффективность работы сотрудников коллекторского отдела.

Поведенческий скоринг (Behavioral scoring) применяется для динамической оценки платежеспособности клиента. С помощью моделей поведенческого скоринга важно предугадывать проблемы у клиента с возвратом кредита и менять его лимиты. Этого можно достичь, используя, например, статистку операций клиента по банковским кредитным картам. Использованию моделей Behavioral scoring посвящено достаточно много трудов, в частности, эти вопросы подробно рассмотрены в [9].

Главным достоинством скоринговых систем является уменьшение кредитных рисков для банка. Кроме этого, скоринговые системы помогают автоматизировать процессы принятия решения и сократить требующееся на это время.

К недостаткам скоринговых систем следует отнести необходимость их постоянной поддержки и модернизации, т.к. такие системы реагируют на меняющуюся экономическую ситуацию с запозданием, поскольку позволяют проводить анализ только данных прошлых клиентов. Скоринг не может точно определить, вернет ли человек заем, но может, оперируя историей прошлых клиентов, провести анализ и выделить, какие заемщики имеют склонность отдавать кредит, а какие - нет. То есть, скоринг проводит параллель между потенциальным заемщиком и прошлыми клиентами по определенным признакам (доход, образование, семейное положение и т.д.) и предполагает, сможет ли первый вернуть кредит. Однако от этого может пострадать порядочный заемщик: если он по характеристикам похож на недобросовестных клиентов, то кредит он вряд ли сможет получить. Для борьбы с этим необходимо применение технологий искусственного интеллекта и машинного обучения в скоринговых системах [19-23].

Классическая технология скоринга

Реализация классической технологии скоринга представлена на рисунке 1.

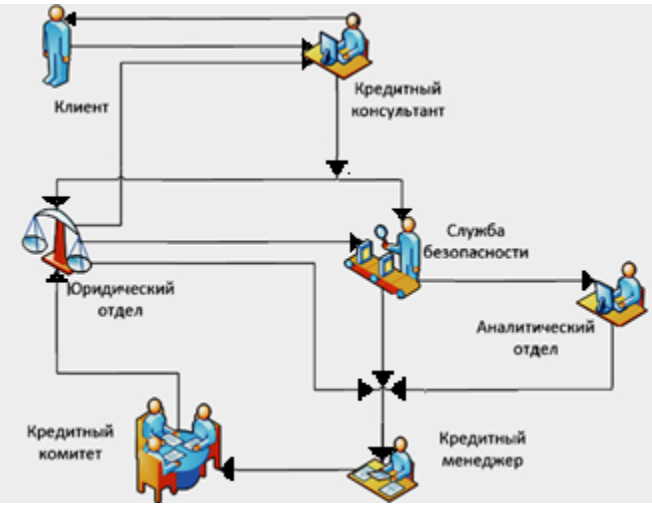


Рис. 1. Классическая технология скоринга

Клиент, обращающийся в банк за получением кредита, подает заявку, где содержатся исходные данные о требуемой ссуде: цель, размер кредита, вид и срок ссуды, предполагаемое обеспечение. На этом этапе с ним работает кредитный консультант, который рассказывает о необходимом пакете документов. Также клиент заполняет анкету со своими данными.

Задачей консультанта является отказать тем потенциальным заемщикам, у которых проявляются проблемы с платежеспособностью. После этого собранный пакет документов консультант отправляет в юридический отдел и службу безопасности.

Юридическое подразделение проверяет документы заемщика на подлинность, после чего делает заключение по клиенту и также передает его в службу безопасности. При наличии у клиента залогового имущества юристы рассматривают возможность его принятия и при положительном решении готовят к рассмотрению соответствующие документы.

Служба безопасности делает запрос в Бюро кредитных историй, где собрана вся информация по кредитам физического или юридического лица, включая микрокредитование, и проверяет достоверность предоставленных клиентом данных. После проведения проверок отдел передает анкету и кредитную историю аналитикам, а заключение юристов поступает к кредитному менеджеру.

Аналитический отдел с помощью оценки кредитного риска определяет максимально возможный лимит займа. Этот лимит устанавливается на основе рейтинговой системы, в основе лежит оценка платежеспособности клиента на основе разработанных в конкретном банке методик. Заключение аналитического отдела также поступает к кредитному менеджеру. Таким образом, на кредитном менеджере замыкаются результаты проверок потенциального заемщика всеми службами банка.

Кредитный менеджер анализирует и обобщает представленные другими подразделениями материалы и готовит заключение для кредитного комитета банка. Кредитный комитет в установленном порядке рассматривает вопрос о предоставлении кредита и принимает соответствующее решение. После этого передает его юридическому отделу.

В случае положительного решения по конкретному клиенту кредитный менеджер информирует его, после чего юридический отдел готовит кредитный договор, который подписывается клиентом и банком. На основании кредитного договора клиенту перечисляются необходимые денежные средства.

Такая технология используется в настоящее время в подавляющем большинстве коммерческих банков. При этом процесс выдачи кредита является достаточно длительным, и, несмотря на это, банки не застрахованы от ошибок. Для снижения вероятности таких ошибок и ускорения самого процесса исследуем возможность внедрения в процесс выдачи кредита методов машинного обучения.

В качестве исходных данных будут использоваться прошлые данные клиентов.

Методы машинного обучения, применяемые в кредитном скоринге

Основная цель машинного обучения при ведении бизнеса состоит в том, чтобы по имеющимся данным построить такую модель, которая может давать такие предсказания, которые повышают его эффективность.

Существует множество методов машинного обучения. В наше время известно много методов машинного обучения. На рисунке 2 представлена общая карта мира машинного обучения.

Задача кредитного скоринга относится к задачам бинарной классификации [10]. Для ее решения можно использовать, например, такие алгоритмы:

- логистическая регрессия (Logistic regression);
- k-ближайших соседей (kNN);
- деревья принятия решений (decision trees);
- случайный лес (random forest);
- нейронные сети.

Рис. 2. Карта мира машинного обучения

Логистическая регрессия представляет собой статистическую модель, которая используется для получения вероятности наступления некоторого события, подгоняя данные к логистической кривой. Для этого анализируются связи между несколькими независимыми переменными (называемыми также регрессорами или предикторами) и зависимой переменной. В кредитном скоринге с помощью логистической регрессии можно ранжировать вероятности возврата заемщиками кредита. Затем в соответствии с политикой банка выбрать интересующих клиентов [11].

Суть метода k-ближайших соседей состоит в отнесении нового элемента к классу, который является самым распространенным среди k-ближайших объектов. В основу данного метода положена т.н. «гипотеза компактности», которая предполагает, что расположенные близко друг к другу объекты в пространстве признаков имеют схожие значения целевой переменной или принадлежат к одному классу. На рисунке 3 приведена графическая иллюстрация работы данного метода.

Рис. 3. Иллюстрация работы метода k-ближайших соседей

Данный алгоритм является достаточно простым, однако к его недостаткам следует отнести необходимость просмотра всех объектов всех классов для принятия окончательного решения. Применению данного метода в банковском скоринге описано, в частности, в работе [12].

Деревья принятия решений представляют собой популярный алгоритм, используемый при классификации некоторых объектов. Данный метод машинного обучения используют, чтобы разделить большой объём входных данных на относительно небольшие группы и прогнозировать наступление события в зависимости от определённых условий. Дерево состоит из «ветвей» – атрибутов, «листьев» – значений целевой функции и узлов – атрибутов, которые отличают случаи. Такое дерево анализирует и принимает решение, как человек. Из его преимуществ можно отметить отсутствие необходимости подготовки данных и интерпретируемости, а из недостатков – переобучаемость и невозможность использования его в чистом виде. Пример работы дерева принятия решений пребдставлен на рисунке 4. Применению данного метода в банковской сфере посвящены работы [13, 14].

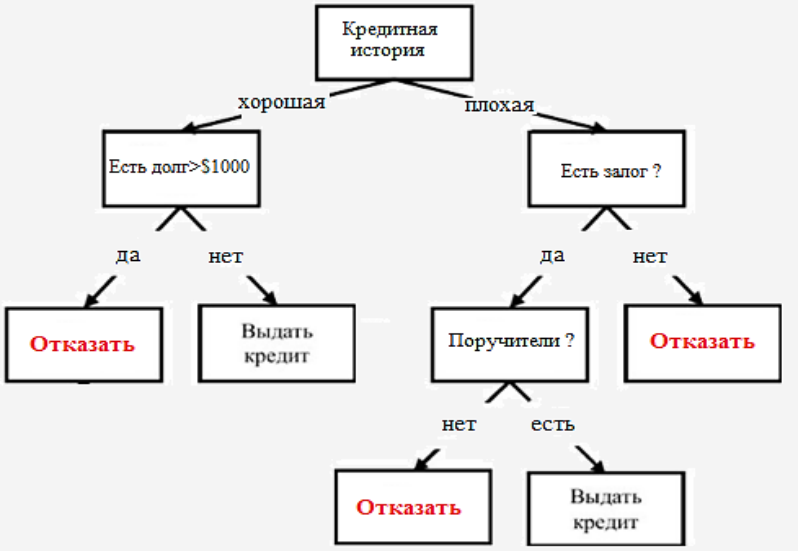


Рис. 4. Пример работы дерева принятия решений

Случайный лес представляет собой комбинацию деревьев принятия решений. Так как одно дерево не может обладать высокой точностью, можно увеличить их, после чего улучшится точность предсказаний. Каждое дерево обучается на своей выборке с элементом случайности. В классификации решение принимается большинством из всех деревьев. Преимущества случайного леса состоит в его универсальности; данный метод используется как в задачах регрессии, так и в задачах классификации. Данный метод машинного обучения не особо чувствителен к выбросам и масштабированию и может обрабатывать большое количество признаков. Из недостатков можно отметить его плохую интерпретируемость и большое потребление памяти (в случае наличия большого числа деревьев). Графическая иллюстрация работы метода случайного леса приведена на рисунке 5. В работе [15] детально рассмотрены возможности применения метода случайного леса для решения задач кредитного скоринга.

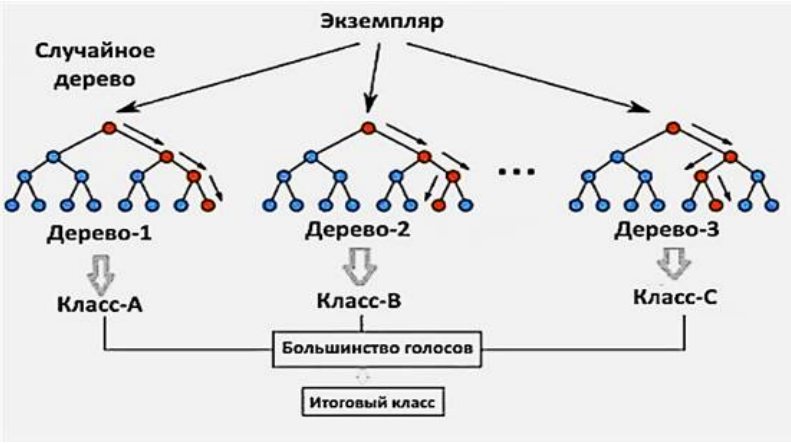


Рис. 5. Пример работы случайного леса

Нейронная (искусственная) сеть является мощным методом, в основе которого лежит работа биологических нейронных сетей. Эта модель состоит из нейронов, объеденных в слои, каждый из которых получает информацию, производит несложные вычисления и передает дальше. Нейронная сеть помогает находить сложные, нетривиальные зависимости. Её недостаток состоит в отсутствии интерпретации и невозможности объяснения предсказания.

Нейронная сеть хорошо справляется с анализом данных, однако если атрибутов много, то для ее создания потребуется внушительное количество вычислительной мощности. На рисунке 6. приведена иллюстрация работы нейронной сети.

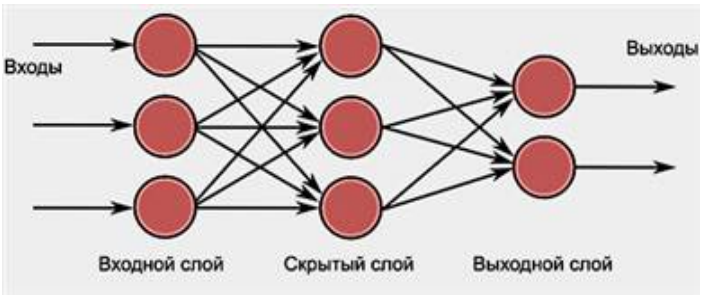


Рис. 6. Работа нейронной сети

Технология решения задачи скоринга с использованием нейронных сетей

Поскольку наибольший объем информации при принятии решения о выдаче или отказе в кредите обрабатывается в аналитическом отделе, то именно в этом отделе представляется целесообразным внедрить методы машинного обучения. Внедрение в отдел аналитики анкетного скоринга на основе машинного обучения повысит скорость обработки заявки и сократит необходимость в большом количестве сотрудников, принимающих решение о выдаче кредита.

Рассмотрим бизнес-процессы реализации этой задачи. Контекстная диаграмма обработки заявки на получение кредита в нотации IDEF0 приведена на рисунке 7, а ее декомпозиция – на рисунке 8.

Рис. 7. Контекстная диаграмма обработки заявки для предоставления кредита в нотации IDEF0

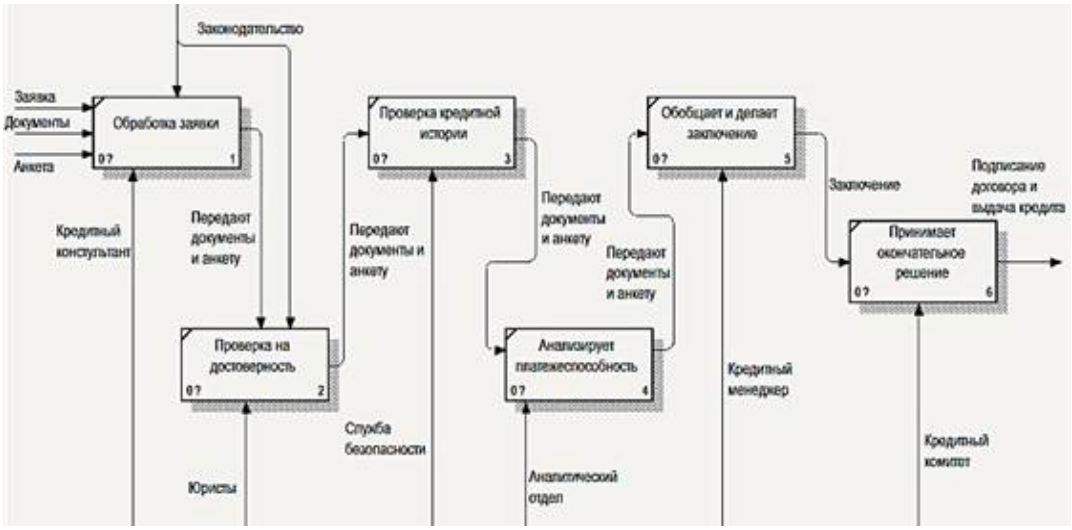


Рис. 8. Декомпозиция первого уровня системы в нотации IDEF0

Кредитный консультант, соблюдая законодательство, обрабатывает заявку клиента и передает собранную информацию далее в юридический отдел. Юристы проверяют на достоверность предоставленные клиентом данные. Служба безопасности проверяет кредитную историю клиента. Аналитический отдел проводит анализ платежеспособности. Кредитный менеджер собирает всю информацию из отделов и обобщает ее для кредитного комитета, который в свою очередь принимает окончательное решение о выдаче кредита.

Следующим этапом решения задачи является описание потоков данных. Для этого используется нотация DFD, которая позволяет разработать модель информационной системы и отобразить процессы обработки, передачи и хранения данных. Контекстная диаграмма обработки заявки для предоставления кредита в нотации DFD представлена на рисунке 9, а ее декомпозиция – на рисунке 10.

Рис. 9. Контекстная диаграмма обработки заявки для предоставления кредита в нотации DFD

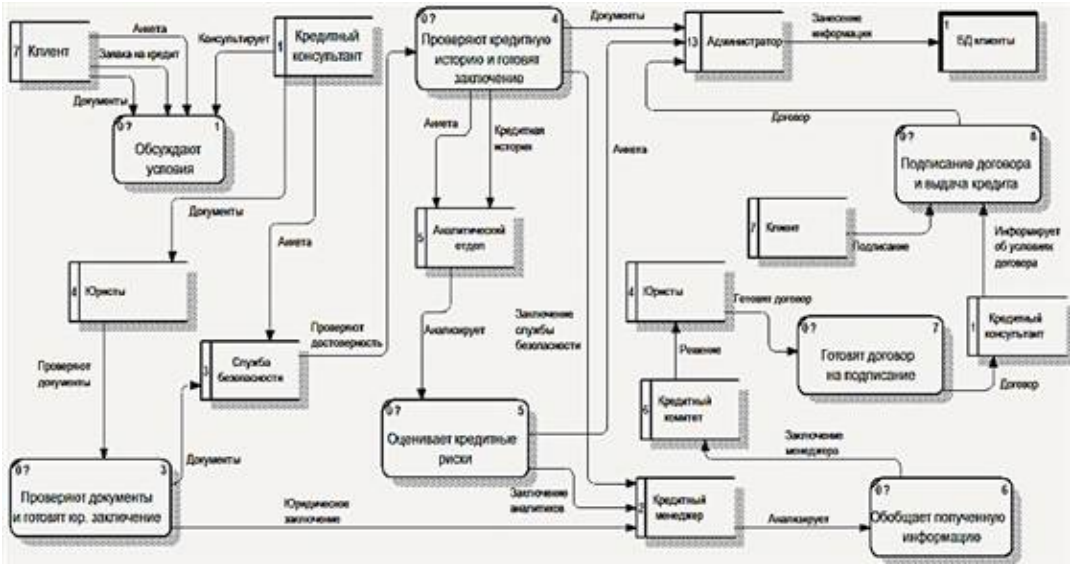


Рис. 10. Декомпозиция первого уровня системы в нотации DFD

Практическая реализация процесса обучения нейронной сети

Рассмотрим процесс применения нейронной сети в аналитическом отделе банка. Аналитик вводит данные нового клиента в интерактивную оболочку Jupiter Notebook [16] и приводит информацию о нем в понятный для машины вид. После этого останется только запустить ранее обученную модель, которая выдаст решение о разрешении или отказе по выдаче кредита.

Для хорошей работы модели необходимо предварительно обработать сырые данные. Их нужно очистить и привести к формату для машинного обучения. Это занимает большую часть времени ана-

литика. В данных бывают пропуски, выбросы, дубликаты, разные регистры и опечатки. Очевидно, что дубликаты, которые не несут ценности для обучения модели, можно удалить, разные регистры и опечатки – найти и исправить. Остальное важно рассмотреть подробнее.

Выбросы могут сильно испортить обучаемую модель. Рассмотрим пример, когда пара клиентов имеет доход в несколько десятков миллионов, а у клиентов остальных не доходит и до 1 миллиона, это явно будет выбросом. Для их нахождения можно использовать квартили – это значения, которые делят объем выборки на четыре части: 25%, 50%, 75%, 100%. Также для количественных признаков можно использовать коробчатую диаграмму (ящик с усами). Так можно понять, является ли клиент выбросом в выборке.

В данных часто бывают пропуски. Это может быть связано как с невнимательностью заполняющего, так и с возражением клиента делиться этими данными. В зависимости от количества данных пропуски можно удалить или заполнить. Незаполненную информацию можно заменить нулями, средним значением, медианой и модой, также можно попробовать использовать машинное обучение. Например, на основе данных о поле клиента можно провести оценку вероятности возврата кредита в зависимости от пола клиента заполнить вычисленным значением пустое значение.

Следующим этапом обучения является кодирование данных кредитным аналитиком. Поскольку большинство алгоритмов машинного обучения обрабатывает только количественные признаки, категориальные переменные следует преобразовать. Для этого можно воспользоваться классом LabelEncoder из популярной библиотеки машинного обучения scikit-learn. Результатом применения данного класса является преобразование каждой категории в числовой признак. В таблице 1 показан пример реализации данного класса, преобразующий город проживания заемщика в числа.

Таблица 1

Иллюстрация работы кодировщика LabelEncoder

	City	Class	Degree	Income	City_le
0	Moscow	A	1	10,2	2
1	London	B	1	11,6	1
2	London	A	2	8,8	1
3	Kiev	A	2	9,0	0
4	Moscow	B	3	6,6	2
5	Moscow	B	3	10,0	2
6	Kiev	A	1	9,0	0
7	Moscow	A	1	7,2	2

Недостатком такой кодировки является создание иерархии данных. В частности, для случая табл. 11, некоторые города будут для модели иметь большие веса перед другими. Для решения этой проблемы возможно использовать класс OneHotEncoder из той же библиотеки. Он создаст столько столбцов, сколько всего городов, и будет проставлять 1 к соответствующим значениям. Теперь в данных вместо одной будет три колонки, где у каждого объекта будет одна единица у столбца с соответствующим городом. Иллюстрация работы данного класса приведена в таблице 2.

Таблица 2.

Иллюстрация работы кодировщика OneHotEncoder

	City	Class	Degree	Income	City=0	City=1	City=2
0	Moscow	A	1	10,2	0	0	1
1	London	B	1	11,6	0	1	0
2	London	A	2	8,8	0	1	0
3	Kiev	A	2	9,0	1	0	0
4	Moscow	B	3	6,6	0	0	1
5	Moscow	B	3	10,0	0	0	1
6	Kiev	A	1	9,0	1	0	0
7	Moscow	A	1	7,2	0	0	1

Следующим этапом реализации процесса обучения модели является нахождения корреляции между различными признаками. Для этого используем коэффициент корреляции Пирсона [17]. С помощью данного коэффициента возможно определить взаимосвязь между двумя переменными. Такая взаимосвязь характеризуется численным значением в диапазоне от -1 до +1. Значение -1 говорит об

отрицательной корреляции, а 1 о положительной взаимосвязи между двумя признаками, их называют также коллинеарными. Если найдется взаимосвязь, то можно удалить некоторые признаки, что положительно повлияет на точность модели.

После определения корреляции закодированные признаки необходимо масштабировать (иногда данный процесс называют шкалированием). Если имеется признак с суммой запрашиваемого кредита, где числа доходят до сотен тысяч, он будет для некоторых алгоритмов машинного обучения (как логистическая регрессия, метод опорных векторов, нейронная сеть) при увеличении иметь больший вес, чем остальные признаки, такие, как возраст, которые не доходят и до ста. То есть при увеличении возраста и кредита на одну единицу, для алгоритма они будут иметь одинаковое значение. Поэтому нужно обязательно нормализовать переменные. Из часто используемых алгоритмов нормализации выделяются нормализация и стандартизация. Нормализация приводит минимальное число в 0, максимальное в 1. Стандартизация подразумевает, что под 0 будет среднее значение, а под 1 дисперсия.

Для того, чтобы можно было посмотреть, не переобучилась ли модель, то есть алгоритм смог найти закономерности и научился предсказывать результат с той или иной точностью, нужно разбить данные на две выборки: обучающую и тестовую. Стандартом является разделение имеющихся данных на 70-80% для обучающей выборки и 20-30% для тестовой. Данные могут быть отсортированы, поэтому какая-то информация может не попасть в одну из выборок. Например, в текстовую переменную не попадут клиенты, не вернувшие заем. Поэтому важно соблюдать случайное распределение данных.

Оценка качества обучения нейронной сети

После построения моделей нужно оценить их качество, выбрать из них лучшую. Для этого нужны метрики для измерения качества предсказаний алгоритма. В задачах бинарной классификации используют такие метрики, как

- Accuracy (аккуратность) – доля правильных ответов алгоритма,
- Precision (точность) – доля объектов, названных классификатором положительными и при этом действительно являющимися положительными),
- Recall (отзыв, или полнтоа) – доля объектов положительного класса из всех объектов положительного класса, найденным алгоритмом),
- F-measure (F1-score) – интегральная метрика, представляющая собой среднее гармоническое между recall и precision.

Для использования таких метрик необходимо понимание матрицы ошибок, используемой для описания вышеперечисленных метрик. Рассмотрим следующий пример. Пусть имеются данные клиентов, котором дали кредит, и они либо отдали его, либо нет. Теперь можно проверить модель на этих данных. В таблице 3 показано сравнение результатов прогноза предсказания модели и реальных результатов по возврату кредита.

Таблица 3

Пример матрицы ошибок

		ПРОГНОЗ	
		YES	**NO**
РЕАЛЬНОСТЬ	**YES**	**TP**	**FN**
	NO	**FP**	**TN**

- TP (True Positive) – это истинно положительные результаты;
- FP (False Positive) – ложно положительные результаты;
- FN (False Negative) – ложно отрицательные результаты;
- TN (True Negative) – истинно отрицательные результаты.

Возможность и необходимость использования той или иной метрики определяется конкретной ситуацией. В соответствии с [17], значение аккуратности может быть определено из выражения:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (1)$$

Допустим, имеется следующая статистика. Среди тысячи заемщиков 950 отдали кредит и 50 не отдали. Если модель будет предсказывать, что все клиенты будут отдавать кредит, то метрика accuracy покажет 95%. Понятно, что такая модель не принесет никакой пользы. В этом случае лучше использовать метрики точности и отзыва. В соответствии с [17] они могут быть определены из выражений:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

В кредитном скоринге под точностью понимается отношение возвращенных кредитов к количеству выданных кредитов. Под отзывом понимается отношение найденных добросовестных заемщиков ко всем клиентам, которые в случае получения средств вернули бы кредит.

Если объединить значение точности и полноты, можно получить интегральное значение, позволяющее найти оптимальный баланс между ними. Для случая одинаковых весов точности и полноты, интегральная оценка вычисляется из выражения [17]:

$$F1 - score = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision} \tag{4}.$$

В случае разных весов точности и полноты, можно использовать выражение:

$$F_\beta = \left(\beta^2 + 1\right) \cdot \frac{Precision \cdot Recall}{\left(\beta^2 \cdot Precision\right) + Recall} \tag{5}$$

При $\beta = 1$ выражение (5) преобразуется к выражению (4). Если же веса у метрик полноты и точности различные, то для случая, когда точность более важна, значение β в выражении (5) принимает значение в интервале [0 …. 1], в противном случае значение β принимает значение, большее 1.

Также можно использовать популярный метод для оценки бинарной классификации AUC ROC (или ROC AUC). Данная метрика определяется путем измерения площади (Area Under Curve) под кривой ошибок (Receiver Operating Characteristic curve). Благодаря ROC-кривой можно посмотреть на модель в целом без введения порогов. Она берет свое начало в точке [0; 0] и доходит до [1; 1], в которой по оси абсцисс – False Positive Rate (ложноположительная частота), а по оси ординат – True Positive Rate (истинноположительная частота) [18]:

$$TRP = \frac{TP}{TP + FN} \tag{6}$$

$$FRP = \frac{FP}{FP + TN} \tag{7}$$

ROC-кривая показывает отношение правильно классифицированных положительных клиентов к неправильно классифицированным отрицательным клиентам.

Пример построения ROC-кривой представлен на рисунке 11. Идеальная модель будет проходить через точку [0; 1] и иметь площадь под кривой, равной 1, а наихудшая – через точку [1; 0] с площадью 0.

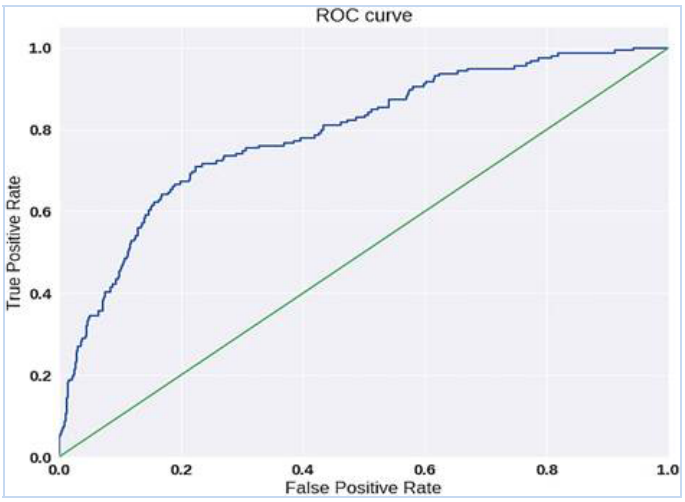


Рис. 11. Пример построения ROC-кривой

Реализация модели анкетного скоринга

Точность предсказательной способности модели зависит от данных, на которых она обучается. Поэтому большую часть времени аналитики работают над данными. Рассмотрим пример, в котором анализируются данные по тысяче клиентов и по 21 признаку. Значения признаков представлены в таблице 4.

Таблица 4

Значения признаков модели анкетного скоринга

default	кредит выдан или отказано
account_check_status	банковский счет
purpose	цель кредита
duration_in_month	на какой срок кредит
credit_history	кредитная история
foreign_worker	иностранец или нет
age	возраст
savings	сберегательный счет
installment_as_income_perc	процентная ставка от дохода
present_emp_since	срок работы на текущей позиции
credit_amount	сумма запрашиваемого кредита
personal_status_sex	пол и семейное положение
present_res_since	место жительства в данный момент
property	имущество
other_installment_plans	другие взносы
housing	имеет ли собственное жилье
credits_this_bank	количество кредитов
job	квалификация работы
people_under_maintenance	количество людей, которых обеспечивает заемщик
telephone	телефон
other_debtors	есть ли поручители

Импортируем необходимые библиотеки (рис. 12).

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

```
%matplotlib inline
plt.rcParams["figure.figsize"] = (12, 9)
```

Рис. 12. Импорт необходимых библиотек для обучения нейронной сети

Выделим из всего набора целевой признак. Таким признаком является признак default, непосредственно отвечающий за выдачу или отказ в выдаче кредита. Запишем этот признак в переменную Y, остальные признаки запишем в переменную X (рис. 13).

```
X = df.drop(['default'], axis = 1)
y = df.default
```

Рис. 13. Выделение целевого признака

Посмотрим, что собой представляют данные:

- df.shape – количество объектов и атрибутов в наборе данных;
- df.info() – признаки, их количество и тип данных;
- df.describe() – количество атрибутов, их среднеквадратическое отклонение, среднее значение, медиану, минимальное и максимальное значения;
- df.nunique() – уникальное количество значений в атрибуте;
- df.isnull().sum() – есть ли пропуски в данных.

В данном наборе данных нет пропусков. Теперь выведем атрибуты, в которых есть значения объектов, составляющих больше 90% данных. Эти столбцы потом можно будет удалить, так как они не принесут пользы для обучаемой модели (рис. 14).

```
num_rows = len(df.index)
low_inf = []

for col in df.columns:
    cnts = df[col].value_counts(dropna=False)
    top_pct = (cnts/num_rows).iloc[0]

    if top_pct > 0.90:
        low_inf.append(col)
        print('{0}: {1:.2f}%'.format(col, top_pct*100))
        print(cnts)
        print()
```

```
other_debtors: 90.70%
none            907
guarantor        52
co-applicant     41
Name: other_debtors, dtype: int64

foreign_worker: 96.30%
yes    963
no      37
Name: foreign_worker, dtype: int64
```

Рис. 14. Выделение доминирующих значений в атрибутах

Обратим внимание на корреляцию между признаками (рис. 15). Есть небольшая корреляция между запросом на кредит и сроком. Эта корреляция в 0.62 не такая большая, поэтому оставим атрибуты без изменений.

```
Corr = X.corr()
sns.heatmap(Corr, annot = True, fmt = '.2f')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x165f66ca448>
```

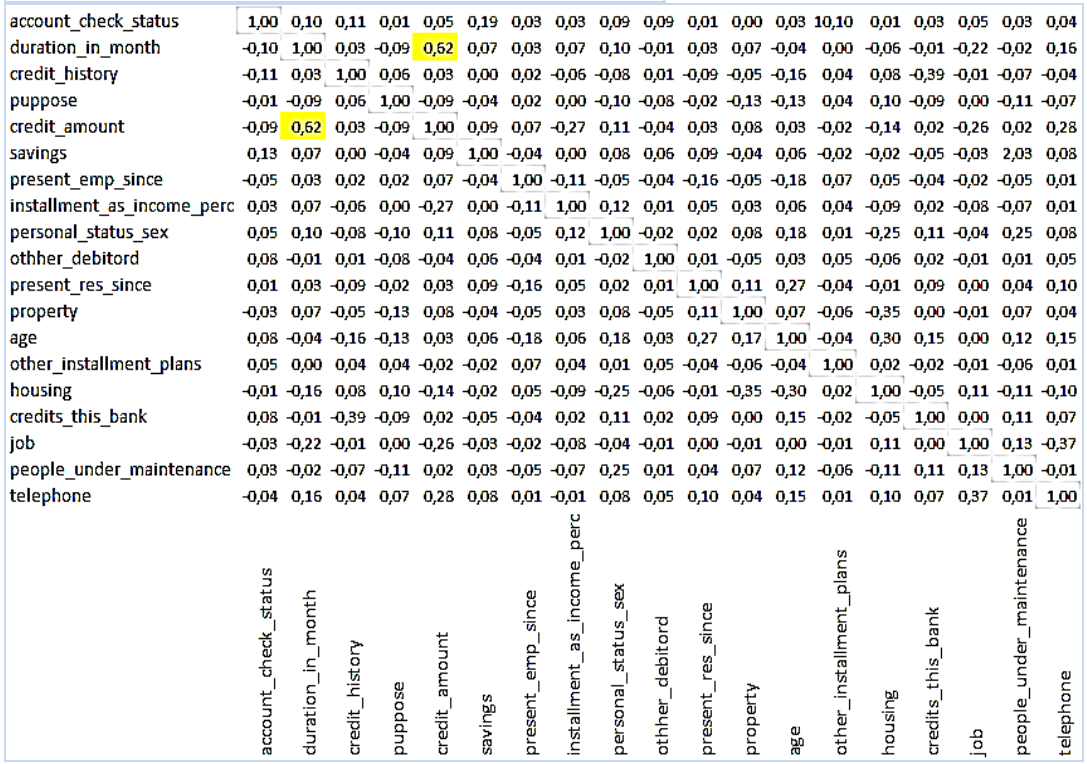


Рис. 15. Корреляции между переменными

Следующим этапом работы является удаление атрибутов, не несущих большой ценности (рис. 16).

```
X.drop(['foreign_worker'], axis = 1, inplace = True)
X.drop(['other_debtors'], axis = 1, inplace = True)
X.drop(['people_under_maintenance'], axis = 1, inplace = True)
X.drop(['credits_this_bank'], axis = 1, inplace = True)
```

Рис. 16. Удаление маловажных атрибутов

С помощью кодировщика LabelEncoder перекодируем категориальные данные (рис. 17).

```
from sklearn.preprocessing import LabelEncoder

LE = LabelEncoder()
X["account_check_status"] = LE.fit_transform(X["account_check_status"])

X["credit_history"] = LE.fit_transform(X["credit_history"])
X["purpose"] = LE.fit_transform(X["purpose"])
X["savings"] = LE.fit_transform(X["savings"])
X["present_emp_since"] = LE.fit_transform(X["present_emp_since"])
X["personal_status_sex"] = LE.fit_transform(X["personal_status_sex"])
X["property"] = LE.fit_transform(X["property"])
X["other_installment_plans"] = LE.fit_transform(X["other_installment_plans"])
X["housing"] = LE.fit_transform(X["housing"])
X["job"] = LE.fit_transform(X["job"])
X["telephone"] = LE.fit_transform(X["telephone"])
```

Рис. 17. Кодирование категориальных признаков

Сформируем выборки для теста и обучения (рис. 18).

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y,
     test_size = 0.2, random_state = 42, stratify = df['default'])

X_train.shape, X_test.shape, y_train.shape, y_test.shape

((800, 16), (200, 16), (800,), (200,))
```

Рис. 18. Формирование выборок

Данные готовы для обучения модели, начнем с алгоритма KNN (рис. 19).

```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=13)

knn.fit(X_train, y_train)

KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=13, p=2,
                     weights='uniform')

knn.score(X_train, y_train)

0.72625

knn.score(X_test, y_test)

0.715
```

Рис. 19. Результат работы алгоритма KNN

Алгоритм показал значение accuracy = 0.715.
Следующим алгоритмом будет дерево решений (рис. 20).

```
from sklearn.model_selection import cross_val_score

for max_depth in max_depth_values:
    clf = tree.DecisionTreeClassifier(criterion = 'entropy', max_depth = max_depth)
    clf.fit(X_train, y_train)
    train_score = clf.score(X_train, y_train)
    test_score = clf.score(X_test, y_test)

    mean_cross_val_score = cross_val_score(clf, X_train, y_train, cv = 5).mean()

    temp_score_df = pd.DataFrame({'max_depth': [max_depth],
                                  'train_score': [train_score],
                                  'test_score': [test_score],
                                  'cross_val_score': [mean_cross_val_score]})
    scores_df = scores_df.append(temp_score_df, sort=False)

scores_df.head(7)
```

	max_depth	train_score	test_score	cross_val_score
0	1	0.70000	0.700	0.70000
0	2	0.70000	0.700	0.69750
0	3	0.73875	0.720	0.69250
0	4	0.77875	0.720	0.71250
0	5	0.79000	0.745	0.73250
0	6	0.81000	0.765	0.70625

Рис. 20. Перебор параметра глубина дерева

Как видно из рисунка 20, при построении ROC-кривой осуществляется перебор значений max_depth, которое отвечает за то, насколько разрастается дерево в глубину. Видно, что при max_depth = 5 достигается наилучшее значение cross_val_score. График ее визуализации показан на рисунке 21.

```
scores_df_long = pd.melt(scores_df, id_vars = ['max_depth'], value_vars =
                         ["train_score", 'test_score','cross_val_score'],
                        var_name = 'set_type', value_name = 'score')
```

```
sns.lineplot(x = 'max_depth', y = 'score', hue = 'set_type', data = scores_df_long)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2450bb4b4c8>
```

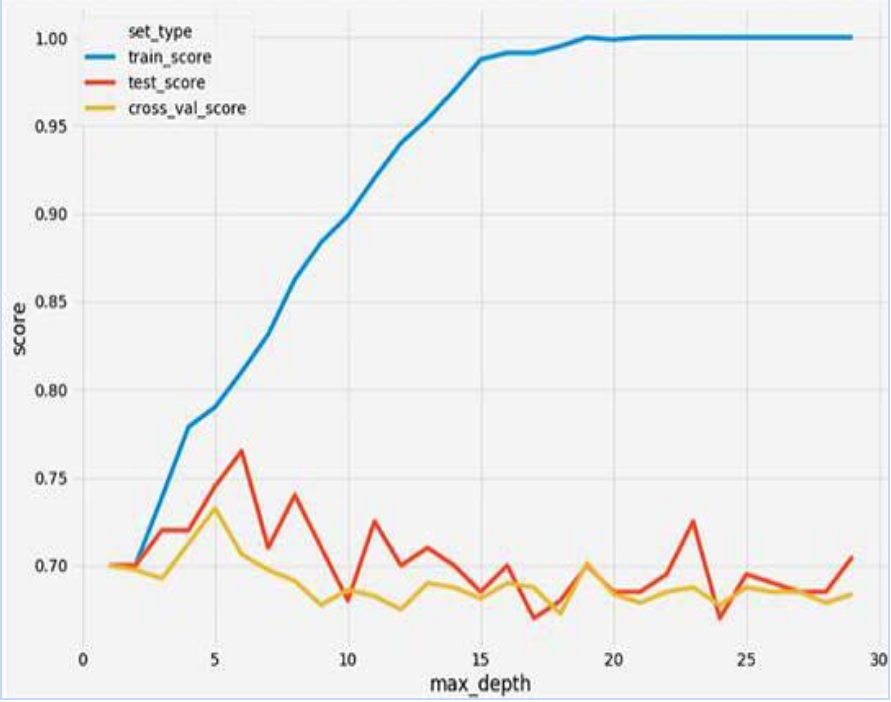


Рис. 21. Зависимость точности алгоритма от глубины дерева

Обучим дерево решений, задав параметр max_depth=5 (рис. 22).

```
best_clf = tree.DecisionTreeClassifier(criterion = 'entropy', max_depth = 5)
```

```
best_clf.fit(X_train, y_train)
```

```
DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=5,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=None, splitter='best')
```

```
best_clf.score(X_test, y_test)
```

```
0.745
```

Рис. 22. Дерево решений при глубине дерева, равной 5

Accuracy дерева решений равен 0.745, что больше, чем у алгоритма KNN.

Благодаря преимуществу дерева в интерпретируемости можно построить визуализацию его обучения (рис. 23):

```
clf = tree.DecisionTreeClassifier(criterion = 'gini', max_depth = 5,
                                                    min_samples_leaf=10)

clf.fit(X_train, y_train)

plt.figure(figsize=(40, 20),dpi=80)
p = tree.plot_tree(clf, fontsize=13,filled=True,feature_names=list(X))
```

Рис. 23. Построение визуализации работы дерева

Перейдем к построению случайного леса (рис. 24).

```
from sklearn.ensemble import RandomForestClassifier

parametrs = {'max_features': [5, 6, 7, 8, 9, 10], 'min_samples_leaf': [1, 2],
             'max_depth': range(10, 15), 'criterion': ['gini', 'entropy']}

rfc = RandomForestClassifier(n_estimators=60, random_state=42, n_jobs=-1,
                                oob_score=True)

gcv= GridSearchCV(rfc, parametrs, n_jobs=-1, cv = 5, verbose=1)

gcv.fit(X_train, y_train)
```

```
gcv.best_params_
```

```
{'criterion': 'entropy',
 'max_depth': 11,
 'max_features': 9,
 'min_samples_leaf': 1}
```

```
best_clf = gcv.best_estimator_
```

```
best_clf.score(X_test, y_test)
```

```
0.8
```

```
y_pred = best_clf.predict(X_test)
```

```
precision_score(y_test, y_pred)
```

```
0.8205128205128205
```

```
recall_score(y_test, y_pred)
```

```
0.9142857142857143
```

Рис. 24. Значения метрик случайного леса

Accuracy случайного леса в 0.8 оказалась максимальной среди рассматриваемых ранее алгоритмов машинного обучения. Эта метрика показывает, что вероятность правильных классификаций скоринга равна 80%. Precision в 82% демонстрирует, сколько кредитов из всех выданных банком будут возвращены, а recall в 91,43% – что модель нашла почти всех клиентов, способных вернуть заем.

Рассмотрим влияние каждого атрибута на обучение модели (рис. 25).

```
feature_importances = best_clf.feature_importances_
feature_importances_df = pd.DataFrame({'features': X_train.columns,
                                       'feature_importances': feature_importances})
feature_importances_df.sort_values('feature_importances', ascending=False)
```

	features	feature_importances
4	credit_amount	0.176357
0	account_check_status	0.132689
11	age	0.116291
1	duration_in_month	0.109974
3	purpose	0.073092
6	present_emp_since	0.049592
9	present_res_since	0.043413
5	savings	0.043226
2	credit_history	0.042919
7	installment_as_income_perc	0.039647
10	property	0.039413
12	other_installment_plans	0.034859
8	personal_status_sex	0.032319
14	job	0.030211
13	housing	0.019177
15	telephone	0.016823

Рис. 25. Влияние атрибутов на обучение модели

Самыми важными признаками оказались сумма запрашиваемого кредита, банковский счет, срок кредита и возраст. Наименьший вклад в обучение внес атрибут телефон. Его можно попробовать убрать и посмотреть, улучшатся ли метрики модели.

Обратимся к распределению вероятностей по выдаче кредита (рис. 26).

```
y_predicted_prob = best_clf.predict_proba(X_test)
pd.Series(y_predicted_prob[:, 1]).hist()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2450f17d348>
```

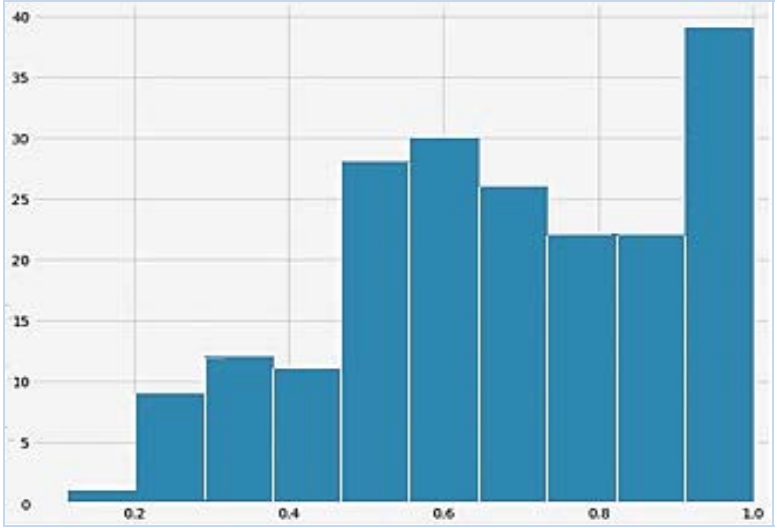


Рис. 26. Вероятности по выдаче кредита для клиентов

Алгоритм считает, что клиенту можно выдать заем, если его вероятность на возврат больше 50%. Если нас не устраивают метрики, например, банк может себе позволить невозврат кредита только в 5%, то тогда можем поменять значение вероятности, используя которую модель выдает кредит (рис. 27).

```
y_pred = np.where(y_predicted_prob[:, 1] > 0.9, 1, 0)
```

```
precision_score(y_test, y_pred)
```

```
0.9523809523809523
```

```
recall_score(y_test, y_pred)
```

```
0.2857142857142857
```

```
accuracy_score(y_test, y_pred)
```

```
0.49
```

Рис. 27. Значения метрик при вероятности принимаемого решения больше 90%

Построим ROC-кривую (рис. 28).

```
from sklearn.metrics import roc_curve, auc
fpr, tpr, thresholds = roc_curve(y_test, y_predicted_prob[:, 1])
roc_auc= auc(fpr, tpr)
plt.figure()
plt.plot(fpr, tpr, label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], "k--")
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic example')
plt.legend(loc="lower right")
plt.show()
```

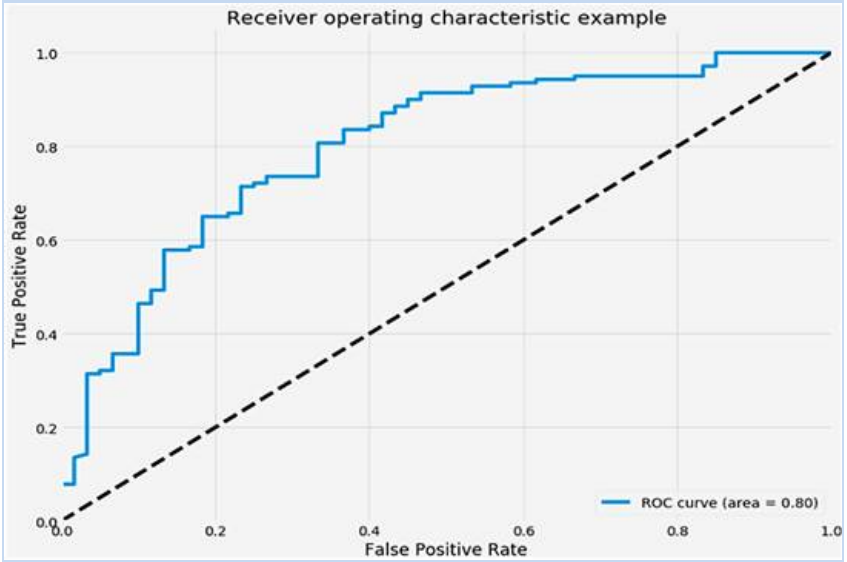


Рис. 28. Roc-кривая алгоритма случайного леса

Значение 0.8 – это отношение TPR к FPR; где TPR – это отношение найденных добросовестных заемщиков ко всем клиентам, которые в случае получения средств вернули бы их, а FPR – отношение ошибочно выданных займов ко всем недобросовестным клиентам, не способных вернуть кредит.

Заключение

Внедрение алгоритмов машинного обучения для решения задачи банковского скоринга позволяет значительно сократить человеческие ресурсы и повысить качество принимаемых решений по выдаче банковских кредитов.

Умелое применение технологий машинного обучения позволяет значительно снизить затраты банков, дать им новые возможности по созданию инновационных банковских продуктов автоматизации бизнес-процессов основной деятельности.

Литература

1. *Ванина М.Ф., Ерохин А.Г., Фролова Е.А.* Скоринг как инновационный инструмент маркетинга // Системы синхронизации, формирования и обработки сигналов, №2-2023. С. 4-12.
2. *Ерохин А.Г., Стуколов С.С., Стуколова А.А.* Роль скоринга в управлении марктинговыми компаниями предприятия // Труды международной научно-технической конференции «Телекоммуникационные и вычислительные системы – 2020». М.: Горячая линия − Телеком, 2020. С. 754-760.
3. *Ванина М.Ф., Ерохин А.Г.* Data Scientist и Machine Learning: интеграция подготовки специалистов IT-направлений и бизнес-технологий // Технологии информационного общества: сб. трудов XIV Междунар. отраслевой науч.-техн. конф. «Технологии информационного общества» (18-19 марта 2020 г., Москва, МТУСИ). М.: ИД «Медиа Паблишер», 2020. С. 558-560.
4. *Ванина М.Ф., Ерохин А.Г.* Повышение эффективности бизнеса компании на основе технологий Big Data и Machine Learning // Технологии информационного общества: сб. трудов XIV Междунар. отраслевой науч.-техн. конф. «Технологии информационного общества» (18-19 марта 2020 г., Москва, МТУСИ). М.: ИД «Медиа Паблишер», 2020. С. 336-338.
5. Банки.ру – финансовый маркетплейс. Вклады, кредиты, ипотека, страховые и инвестиционные продукты [Электронный ресурс]. URL: https://www.banki.ru/ (дата обращения 20.01.2025).
6. *Банных А.А., Летчиков А.А.* Методика оценки кредитного риска заемщика с применением скоринга бюро кредитных историй // Вестник Удмурдского университета. Серия Экономика и право, 2013, вып. 4. С. 5-9.
7. *Laleh, Naeimeh & Abdollahi Azgomi, Mohammad*. (2010). A hybrid fraud scoring and spike detection technique in streaming data. Intell. Data Anal. 14. 773-800. 10.3233/IDA-2010-0451.
8. *Ismagilov, Ilyas & Sabirova, Ajgul & Kataseva, Dina & Katasev, Alexey*. (2021). Collection Scoring Models Development and Research Based on the Deductor Analytical Platform. Nexo Revista Científica. 33. 608-615. 10.5377/nexo.v33i02.10796.
9. *Nikolaidis, Jim & Doumpos, Michael & Zopounidis, Constantin*. (2015). Exploring Population Drift on Behavioral Scoring. 5th International Conference of the Financial Engineering and Banking Society. At: Nantes, France.
10. *Кисляков А.Н.* Алгоритм бинарной классификации на основе графов принятия решений в задачах кредитного скоринга // Модели, системы, сети в экономике, технике, природе и обществе. 2021. № 1. С. 29-41. Doi:10.21685-8486-2021-1-3.
11. *Сорокин А.С.* Построение скоринговых карт с использованием модели логистической регрессии // Интернет-журнал «Науковоедение», Выпуск 2, март-апрель 2014 [Электронный ресурс], URL: https://naukovedenie.ru/PDF/180EVN214.pdf (дата обращения 20.01.2025). 29 с.
12. *Алексеева В.А., Калимуллина Р.И.* Применение метода ближайших соседей при моделировании кредитных рисков // Вестник УлГТУ, № 3/2014. С. 54-56.
13. *Груздев А.* Разработка скоринговой модели с помощью метода деревьев решений // Риск-менеджмент в кредитной организации, №2/2013.
14. *Стадников А.О., Белоусов А.А.* (науч. рук.). Сравнение моделей кредитного скоринга на базе методов решающих деревьев // Международный научный журнал «Инновационная наука», № 6-1, 2022. С. 46-50.
15. *Груздев А.* Метод случайного леса в скоринге // Риск-менеджмент в кредитной организации, №1/2014.
16. *Toomey D*. Jupyter for Data Science, Packt Publishing, ISBN: 9781785880070, 2017.
17. *Гржибовский А.М.* Корреляционный анализ // Экология человека, 2008, № 9. С. 50-60.
18. Метрики классификации и регрессии [Электронный ресурс], URL: https://education.yandex.ru/handbook/ml/article/metriki-klassifikacii-i-regressii (дата обращения 20.01.2025).
19. How to explain the ROC curve and ROC AUC score? [Электронный ресурс], URL: https://www.evidentlyai.com/classification-metrics/explain-roc-curve (Дата обращения 20.01.2025).
20. *Kirov D.E., Toutova N.V., Vorozhtsov A.S., Andreev I.A.* Feature selection for predicting live migration characteristics of virtual machines // T-Comm. 2021. Т. 15. № 7. С. 62-70. EDN: AGGBDW
21. *Тутов А.В., Тутова Н.В., Ворожцов А.С., Андреев И.А.* Многокритериальная оптимизация размещения виртуальных машин по физическим серверам в облачных центрах обработки данных // T-Comm: Телекоммуникации и транспорт. 2021. Т. 15. № 1. С. 28-34. EDN: IOFQSS
22. *Губин А.С., Тутова Н.В.* Анализ подхода к разработке приложений с "чистой" архитектурой // Телекоммуникации и информационные технологии. 2022. Т. 9. № 1. С. 28-37. EDN: NOZMKG
23. *Дубельщиков А.А., Тутова Н.В.* Навыки яндекс.алиса: от идеи до реализации // Телекоммуникации и информационные технологии. 2020. Т. 7. № 2. С. 92-97. EDN: ZIFDYG

СРАВНЕНИЕ ТЕКСТОВЫХ НАБОРОВ ДАННЫХ ПРИ ПОМОЩИ КОСИНУСНОГО СХОДСТВА

Гадасин Денис Вадимович
МТУСИ, заместитель заведующего кафедрой СИТиС, к.т.н., доцент, Москва, Россия,
d.v.gadasin@mtuci.ru

Чернышов Дмитрий Викторович
МТУСИ, магистрант, Москва, Россия
chernyshov_d.v@mail.ru

Комкова Марина Георгиевна
МТУСИ, старший преподаватель кафедры СИТиС, Москва, Россия,
d.v.gadasin@mtuci.ru

Михайлов Михаил Романович
магистрант МТУСИ, Москва, Россия,
mishamikhaylov2019@mail.ru

Аннотация
В работе проводится практический эксперимент по процессу подготовки текстового набора данных и сравнения текстовых наборов с помощью косинусного сходства. В работе также представлен процесс нахождения ключевых слов и их частоты, с последующим составлением векторов текстовых наборов для оценки косинусного сходства. Помимо этого, в работе оценивается влияние количества ключевых слов на результат косинусного сходства, а также рассматривается применение биграмм для оценки схожести текстовых наборов данных.

Ключевые слова
Биграммы, косинусное сходство, частота слов, закон Ципфа, естественность языка, ключевые слова, лемматизация, структурирование

Введение

Задача структурирования текстовых данных связана с стремительным ростом объема текстовой информации в сети интернет [22-29]. Эффективная обработка и анализ данных позволит получить необходимую информации о содержании текстового набора данных. Ручная обработка большого объема информации является трудоемким процессом и несет риски, связанные с человеческим фактором. В связи с этим необходимо автоматизировать процесс обработки и структурирования текстовых данных [1-3]. Автоматизация значительно сократит время анализа текста и позволит применить дополнительные методы для повышения точности классификации данных и улучшит процесс проведения структурирования.

Дополнительные методы, которые могут применяться для анализа и структурирования, представляют собой проведение частотного анализа и построение дерева предметной области для определения контекста текстового набора данных и его принадлежности к предметной области [4].

Для оценки принадлежности текста относительно других предметных областей может применяться метод анализа иерархий. Данный подход позволит оценить, насколько текстовый набор данных подходит к определенной предметной области.

В данной статье будут рассматриваться процесс оценки релевантности текста на основе ключевых слов и фраз и процесс подготовки к построению дерева предметной области на основе ранее выявленных ключевых слов.

Построение дерева на основе ранее определенных ключевых слов

Перед процессом построения дерева с использованием ранее определенных ключевых слов следует выделить ключевой этап в данном процессе – это поиск ключевых слов [5-7].

Поиск ключевых слов, фраз, которые отражают содержание текстового набора данных является началом в процессе построения дерево. Нахождение слов производится при помощи частотного анализа, который направлен на поиск наиболее часто встречающихся слов в тексте. Иными словами, ча-

стотный анализ выявляет слова, которые преобладают в тексте, а после сортирует их в порядке убывания. Сортировка позволяет отразить характер текстового набора, показывая основную тематику, переходящая в более общую.

Далее полученная сортировка также позволяет определить место ключевого слова в иерархии дерева предметной области. В главном узле дерева располагается наиболее общий термин, например, «финансы», далее от этого узла будет следовать более конкретная категория, например, «банковское дело», следующим узлом уже может быть «накопление капиталов». Такая иерархия позволяет организовать информацию и повысить точность классификации текста [8].

В общем формате дерево предметной области можно представить, как граф, в котором вершины представляют собой ключевые слова, тогда ребра будут отражать связи между узлами, например, отношение к более общей категории. В таком случае, значение веса рассчитывается на основе частоты появления слова и значимости для данной предметной области. Вес каждого узла можно рассчитывать на основе частоты его появления в текстах или его значимости для данной предметной области.

Таким образом, данный подход позволит анализировать новые тексты с помощью сопоставления ключевых самого текста с деревом предметной области. Результатом будет оценка релевантности текста по отношению к предметной области.

При нахождении новых ключевых слов в определенной предметной области необходимо дополнить дерево этой предметной области новыми ключевыми словами. Добавление повысит точность процесса оценки релевантности текста к предметной области.

Для построения дерева будут использоваться ассоциативно иерархический портрет, который является сборником лингвистических знаний, свойственных предметной области. Ассоциативно – иерархический портрет — это совокупность наиболее характерных предметных и лингвистических знаний, свойственных определенной предметной области [9-11] Для выделения значимых слов или терминов предлагается использовать методики их автоматического выявления. Такой методикой как раз будет являться частотный анализ, который позволяет провести статистическое ранжирование и определить слова или лексические последовательности, имеющие самостоятельную значимость. Такие лексические последовательности определяются по абсолютной частоте их встречаемости в тексте предметной области [12].

Метод построения дерева можно предложить в виде следующего алгоритма:

1. Проведение предобработки текста, включающую в себя удаление шума, нормализацию, токенизацию;
2. Выбор ключевых терминов или слов при помощи частотного анализа текста, а также определение естественности текстового набора при помощи закона Ципфа;
3. Построение частотного словаря предметной области из ключевых терминов;
4. Определение внутренних узлов, отражающие тематические категории в рамках данной предметной области;
5. Выбор ключевых слов в состав внутренних узлов.

Дерево предметной области можно рассматривать как направленный граф G = (V, E), где вершины (V) отражают сущности реального мира, а рёбра (E) обозначают отношения между этими сущностями. Таким образом, дерево предметной области описывает обнаружение иерархических отношений между ключевыми понятиями и категориями в тексте [13].

На основе этого можно сказать, что дерево представляет собой обнаружение в тексте сущностей и отношений между ними. Тогда процесс определения внутренних узлов, отражающих тематические категории предметной области, можно описать как процесс определения сущностей и взаимосвязи между ними. Подводя итоги, можно сказать, что использование дерева может использоваться для организации и структурирования текстовой информации. Дерево может использоваться для процесса классификации текстовых наборов данных, помогая определить тематику текста. Добавление дерева новыми ключевыми словами позволит улучшить точность анализа.

Оценка схожести текстов на основе косинусного сходства ключевых слов

Алгоритм оценки релевантности текста к предметной области начинается с анализа ключевых слов и фраз, определенных для каждой предметной области. Для оценки важности каждого ключевого слова используется частотный анализ, учитывающий не только частоту встречаемости, но и контекст. Каждое ключевое слово или фраза может быть оценено по нескольким критериям: частота,

важность в данной предметной области, и контекстное использование.

Пример:

1. Частотный анализ: определяются наиболее часто встречающиеся ключевые слова в тексте, а затем сравниваются с частотами слов, характерных для предметной области;
2. Контекстная релевантность: анализируются биграммы для определения контекста использования ключевых слов, что позволяет выделить слова, которые чаще всего встречаются в определенном контексте, связанном с предметной областью [14, 15].

Биграммы – это последовательности из двух и трех слов соответственно, которые заранее можно определить при помощи токенизации.

Биграммы: состоит из двух соседних слов, например, «базовая инфляция», «валютный контроль».

Для оценки релевантности и важности ключевых слов и фраз предметной области была разработана программа, которая проводит частотный анализ текста и работает по следующему алгоритму (рис. 1).

Рис. 1. Алгоритм оценки релевантности ключевых слов

Заранее был определен словарь, содержащий стоп слова, которые не следует учитывать в частотном анализе, также была проведена нормализация и токенизация [16].

Для оценки текстового набора данных, необходимо выявить ключевые слова и рассчитать их частоту в тексте. После этого можно определить частоту ключевых слов по закону Ципфа для оценки естественности текстового набора данных.

Полученные результаты можно представить в формате вектора, который будет содержать частоты слов. Данные ключевые слова можно интерпретировать как признаки, по которым будет оцениваться схожесть текстов [17].

Для оценки схожести текстовых наборов предлагается воспользоваться метрикой оценки равенства векторов – косинусным сходством векторов (1) [18]. Данная метрика вычисляет угол между векторами и если он близок к 1, то можно говорить о схожести текстовых наборов.

$$\cos(\theta) = \frac{\sum_{i=1}^{n} A_i \cdot B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \cdot \sqrt{\sum_{i=1}^{n} (B_i)^2}} \quad (1)$$

Проведем практический эксперимент. Для определения схожести текстовых наборов данных предлагается использовать рассказ Пришвина М. М. «Лесная капель» и рассказ Пришвина М. М. «За волшебным колобком».

Для получения более точных результатов оценки равенства векторов был добавлен еще один шаг нормализации текста – приведение слов к лемме. Лемматизация позволяет привести слова к начальной (словарной) форме и исключить повторяющиеся слова, которые имеют одинаковый корень, но различные окончания [19]. После лемматизации следуют приведение слов к нижнему регистру, удаление специальных символов и разделение слов на пробелы.

Лемматизация проводилась при помощи при помощи программы MyStem, которая позволяет провести морфологический разбор слов русского языка.

После проведения вышеперечисленных шагов следует определение ключевых слов и нахождение

их частот в тексте, а также формирование двух векторов в формате ключ – значение, где ключом является слово, а значением его частота. Таким образом формируются 2 вектора, которые используются в оценке равенства векторов при вычислении косинусного сходства [20].

Для получения точных результатов равенства векторов было произведено сравнение результатов косинусного сходства при различном количестве ключевых слов. Первый практический пример, представленный на рисунке 2, состоял из вычисления сходства на основе 10 ключевых слов.

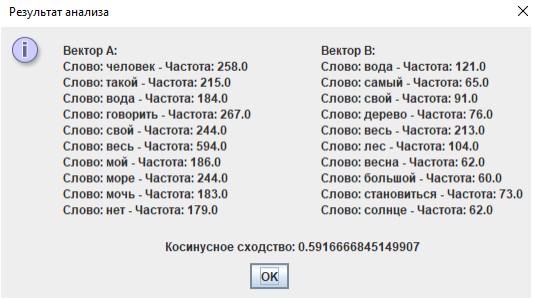


Рис. 2. Первый практический пример

Из результата видно, что косинусное сходство равно примерно 0,59 при количестве 10 ключевых слов. Рассмотрим следующий пример с 100 ключевых слов. На рисунке 3 будет представлен вектор A, а на рисунке 4 вектор B и значение косинусного сходства.

Рис. 3. Вектор A

Из результата можно заметить, что значение косинусного сходства при 100 ключевых слов выросло на 0,05. Для нахождения оптимального значения, при котором косинусное сходство не поменяется в значительном количестве необходимо рассмотреть еще один практический пример (рис. 5).

Из результата видно, что косинусное сходство выросло на 0,04 при 300 ключевых слов.

Последний практический пример представлен на рисунке 6.

Четвертый практический пример был выполнен с 310 ключевыми словами. Из результата можно заметить, что косинусное сходство стало хуже, так как результат стал меньше на 0,012.

Подводя итоги практических примеров, можно сделать вывод, что для более быстрых расчетов с меньшей точностью допускается 0,1% ключевых слов от общего количества слов (определенных после процессов нормализации). Для более точного результата предлагается использовать как минимум 1% ключевых слов от общего количества слов.

Оценивая общее влияние количества ключевых слов на косинусное сходство, можно заметить, что результаты первого и последнего примера расходятся на 0.09, что является незначительным числом при оценке векторов на косинусное сходство [21].

Рис. 4. Вектор B и косинусное сходство

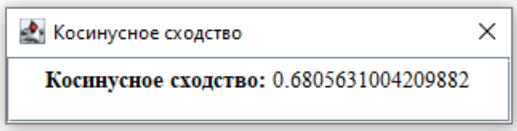


Рис. 5. Третий практический пример

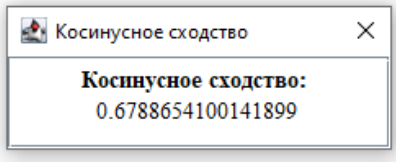


Рис. 6. Четвертый практический пример

Полученный процент естественности, который был рассчитан по формуле 2, составляет 40,42% для рассказа «Лесная капель» и 76,04 для рассказа «За волшебным колобком».

$$w = 100 - \left(\frac{\sum \lambda}{\sum F} \right) \cdot 100 , \tag{2}$$

где w – процент естественности, $\sum \lambda$ – сумма отклонений, $\sum F$ – сумма фактических частот.

Отклонение рассчитывается по формуле 3:

$$\lambda = \left| C - f(r) \right| , \tag{3}$$

где λ – отклонение, C – частота слова и $f(r)$ – частота по закону Ципфа

Частота слова по закону Ципфа рассчитывается по формуле 4:

$$f(r) = \frac{c}{r} , \tag{4}$$

где f (r) – функция распределения, r – ранг слова, C – частота слова.

Фактическая частота слова рассчитывается по формуле 5:

$$f(w) = \frac{n}{N}, \quad (5)$$

где $f(w)$ – частота слова w, n – количество вхождений слова w в текст, N – общее количество слов в тексте.

Помимо вычисления косинусного сходства векторов ключевых слов, было также проведено вычисление косинусного сходства векторов биграмм. Было также выполнено несколько попыток оценки влияния количества ключевых пар на значение косинусного сходства. Максимальное значение, которое удалось достичь было равно 0,077. Исходя из этого можно сказать, что использование биграмм не представляет собой точный способ оценки схожести двух текстовых наборов данных.

Заключение

В ходе исследования был проведен частотный анализ текста, который помог выявить наиболее ключевые слова, необходимые для построения дерева предметной области и определения общего узла дерева. Рост информации в современном мире требует новых методов анализа и структурирования текстовых данных. Одним из ключевых процессов анализа текста является частотный анализ, который позволяет выделить набор слов необходимый для построения дерева. Для структурирования найденных ключевых слов предлагается использовать дерево предметной области, который будет отражать контекст предметной области для характерного текстового набора данных. Дальнейшие подход с применением дерева предметной области поможет обеспечить не только простой процесс классификации текста, но и процесс поиска и структурирования.

В последующих исследованиях данной тематики следует сосредоточиться на исследовании методов, которые позволят выстроить автоматизированный процесс построения дерева, так как на данный момент необходимо составлять частотный словарь, который определяется частотным анализом.

Литература

1. *Золотарева П.Ю., Гадасин Д.В., Маклачков К.А*. Методы обработки информации в распределенных информационных системах // Тенденции развития Интернет и цифровой экономики : Труды VI Международной научно-практической конференции, Симферополь-Алушта, 01-03 июня 2023 г. Симферополь: ИП Зуева, 2023. С. 187-189. EDN LGONZK

2. *Гадасин, Д.В., Вакурин И.С., Тремасова Л.А*. Алгоритм распределения данных между системами хранения на основе свойства самоподобия // Электросвязь. 2024. № 4. С. 44-50. DOI 10.34832/ELSV.2024.53.4.015. EDN BRSLCL

3. *Гадасин Д.В., Шведов А.В.* Применение транспортной задачи для балансировки нагрузки в условиях нечеткости исходных данных // T-Comm: Телекоммуникации и транспорт. 2024. Т. 18, № 1. С. 13-20. DOI 10.36724/2072-8735-2024-18-1-13-20. EDN WKNPIX

4. *Гадасин Д.В., Бессолицын А.Д*. Виды и методы структурирования данных из различных информационных систем: анализ и применение // Актуальные проблемы и перспективы развития экономики, Симферополь – Гурзуф, 12-14 октября 2023 г. Симферополь: ИП Зуева Т. В., 2023. С. 202-204. EDN UGZRXL

5. *Гадасин Д.В., Михайлов М.Р., Чернышов Д.В.* Определение алгоритма структурирования текстовых данных // REDS: Телекоммуникационные устройства и системы. 2024. Т. 14, № 1. С. 4-11. EDN GLAEQF

6. *Гадасин Д.В*. Построение бинарного дерева минимальной цены // T-Comm: Телекоммуникации и транспорт. 2024. Т. 18, № 11. С. 38-44. DOI 10.36724/2072-8735-2024-18-11-38-44. EDN GMCEWG

7. *Гадасин Д.В*., *Тремасова Л.А., Гадасин Д.Д.* Распределение поступающей нагрузки с применением SS-метода // I-methods. 2023. Т. 15, № 3. EDN HQEYTW

8. *Гадасин Д.В., Бессолицын А.Д., Гадасин Д.Д.* Оценка качества данных информационных систем // DSPA: Вопросы применения цифровой обработки сигналов. 2024. Т. 14, № 2. С. 4-12. EDN GYIWJU

9. *Шарнин М.М., Клименко С.В., Хакимова А.Х., Сюракшина Ю.В*. Методика построения ассоциативно-иерархического портрета предметной области. // Сборник трудов Международной конференции CPT1617, 07-14 мая 2017, Ларнака, Кипр. Изд-во Института физико-технической информатики (ИФТИ). 2017. С. 343-349.

10. *Шарнин М.М., Кузнецов И.П.* Автоматическое формирование электронных энциклопедий и справочных пособий по информации из сети «Интернет» // Системы и средства информатики. Вып.14, ИПИ РАН, 2004. С. 210-223.

11. *Шарнин М.М., Сомин Н.В., Кузнецов И.П., Морозова Ю.И., Галина И.В., Козеренко Е.Б*. Статистические механизмы формирования ассоциативных портретов предметных областей на основе естественно-языковых текстов больших объемов для систем извлечения знаний // Информатика и её применения. 2013. Т.7. № 2. С. 92-99.

12. *Клименко С.В., Шарнин М.М., Хакимова А.Х.* и др. Методика построения ассоциативно-иерархического портрета предметной области: иерархия категорий // Ситуационные центры и информационно-аналитические системы класса 4i для задач мониторинга и безопасности (SCVRT2017): Труды Международной научной конференции, Москва – Протвино, 28-30 ноября 2017 г. Москва – Протвино: Автономная некоммерческая организация "Институт физико-технической информатики", 2017. С. 251-260.

13. *Allemang D., Hendler J.* Semantic Web for the Working Ontologist: Effective Modeling for Linked Data, RDFS, and OWL. 2nd ed., Morgan Kaufmann, 2011.

14. *Гадасин Д.В., Шведов А.В., Вакурин И.С., Тремасова Л.А*. Семантический и вероятностный векторы в поисковых запросах // REDS: Телекоммуникационные устройства и системы. 2023. Т. 13, № 2. С. 19-32. EDN HUBVHI

15. *Panteleeva K.A., Gadasin D.D., Shvedov A.V., Gadasin D.V*. Determining the Amount of Information in One Information Bit of Text Data // Systems of Signals Generating and Processing in the Field of on Board Communications. 2023. Vol. 6, No. 1, pp. 360-364. DOI 10.1109/IEEECONF56737.2023.10091972. EDN KFMNSI

16. *Гадасин Д.В., Шведов А.В., Пантелеева К.А*. Предобработка информации для систем машинного обучения // Актуальные проблемы и перспективы развития экономики : Труды XXI Международной научно-практической конференции, Симферополь-Гурзуф, 20-22 октября 2022 года. Симферополь: Крымский федеральный университет им. В.И. Вернадского, 2022. С. 268-269. EDN QVIOMF

17. *Шульпина П.Д., Гадасин Д.В., Тремасова Л.А*. Взвешивание признаков как Предварительная обработка исходных наборов данных // Системы синхронизации, формирования и обработки сигналов. 2024. Т. 15, № 3. С. 40-47. EDN BIOWRB

18. *Гадасин Д.В., Шведов А.В., Вакурин И.С.* Определение семантической близости текстов с использованием алгоритма сравнения сущности графов // REDS: Телекоммуникационные устройства и системы. 2022. Т. 12, № 4. С. 11-19. EDN PVJKQJ

19. *Гадасин Д.В., Пак Е.В., Коровушкина В.М., Мелькова Е.К*. Предобработка текстовой информации на основе термов естественного языка // REDS: Телекоммуникационные устройства и системы. 2022. Т. 12, № 1. С. 4-11. EDN PDGAVP

20. *Shevelev S.V., Shvedov A.V., Gadasin D.V., Vakurin I.S*. Syntax and probability vectors in search query // Wave Electronics and Its Application in Information and Telecommunication Systems. 2023. Vol. 6, No. 1, pp. 407-414. EDN TVFKOH

21. *Гадасин Д.В., Шведов А.В., Пантелеева К.А., Гадасин Д.Д*. Определение порога количества информации для возможности структурирования данных // Телекоммуникационные и вычислительные системы : Юбилейный сборник трудов тридцатого международного научно-технического форума, Москва, 12-15 декабря 2022 года. М.: Издательство МБА, 2022. С. 125-130. EDN MYMHUP

22. *Гадасин Д.В., Шведов А.В., Кузин И.А.* Трехмерная реконструкции объекта по одному изображению с использованием глубоких свёрточных нейронных сетей // T-Comm: Телекоммуникации и транспорт. 2022. Т. 16, № 7. С. 29-35. DOI: 10.36724/2072-8735-2022-16-7-29-35 EDN: YTLCNW

23. *Shvedov A.V., Gadasin D.V., Alyoshintsev A.V.* Segment routing in data transmission networks // T-Comm. 2022. Vol. 16. No. 5, pp. 56-62. DOI: 10.36724/2072-8735-2022-16-5-56-62 EDN: VAYLJQ

24. *Назаров М.Д., Шведов А.В.* Корреляция атрибутов соглашения об уровне обслуживания с основными параметрами QoS в корпоративных сетях // Телекоммуникации и информационные технологии. 2020. Т. 7. № 2. С. 73-79. EDN: VQHDTJ

25. *Kalmykov N.S., Dokuchaev V.A.* Segment routing as a basis for software defined network // T-Comm. 2021. Т. 15. № 7. С. 50-54. EDN: LYVZCV

26. *Dokuchaev V.A., Maklachkova V.V., Statev V.Yu.* Classification of personal data security threats in information systems // T-Comm. 2020. Т. 14. № 1. С. 56-60. EDN: QOGYHH

27. *Докучаев В.А., Маклачкова В.В., Статьев В.Ю.* Цифровизация субъекта персональных данных // T-Comm: Телекоммуникации и транспорт. 2020. Т. 14. № 6. С. 27-32. EDN: XVWYJP

28. *Pavlov S.V., Dokuchaev V.A., Mytenkov S.S.* Model of a fuzzy dynamic decision support system // T-Comm. 2020. Т. 14. № 9. С. 43-47. EDN: VYFNLB

29. Кузин И.А., Гадасин Д.В Модель контейнера данных для минимизации трафика при передаче субъективных характеристик объектов на изображении трехмерной сцены // Телекоммуникации и информационные технологии. 2021. Т. 8. № 2. С. 96-100. EDN: TYFFBH

АНАЛИЗ СПОСОБОВ ПРИМЕНЕНИЯ МОДЕЛЕЙ ГЛУБОКОГО ОБУЧЕНИЯ В ВИДЕОКОМПРЕССИИ

Вотяков Семён Юрьевич
Московский Технический Университет Связи и Информатики, Москва, Россия
s.u.votyakov@mtuci.ru

Комина Анастасия Олеговна
Московский Технический Университет Связи и Информатики, Москва, Россия

Власюк Игорь Викторович
Московский Технический Университет Связи и Информатики, доцент, к.т.н., Москва, Россия

Аннотация
Алгоритмы видеокомпрессии позволяют многократно уменьшать объём видеоинформации. На текущий момент традиционные приёмы сжатия видео исчерпали себя, в связи с этим появляются новые решения. В этой статье проведено исследование развития алгоритмов кодирования видео. Наряду с классическими стандартами и методами, в представленном материале приведены решения с использованием машинного обучения. Предложен алгоритм с реставрационной обработкой на основе машинного обучения в петле обратной связи.

Ключевые слова
Алгоритмы видеокомпрессии, машинное обучение, глубокое обучение, обработка сигналов, реставрационная обработка

Введение

В 2022 году на видео приходится 82% от всего интернет-трафика [1]. В нынешних условиях эффективное сжатие видеоданных имеет решающее значение для компаний, занимающихся потоковой передачей видео. На современном этапе развития технологии системы кодирования видео показывают удовлетворительные результаты, однако темпы роста эффективности алгоритмов сжатия не соответствуют ожиданиям [10-17].

За последние три десятилетия, начиная с кодека MPEG-2 [2] прослеживается тенденция в улучшении производительности видеокодеков, подобная закону Мура в области изготовления интегральных микросхем. Согласно наблюдению, сделанному профессором Домански [3], приблизительно каждые девять лет происходит увеличение эффективности кодеков вдвое. Так, AVC позволяет кодировать тот же контент что и MPEG-2 с вдвое меньшим битрейтом. В свою очередь с такими же относительными показателями HEVC [4] превосходил AVC [5].

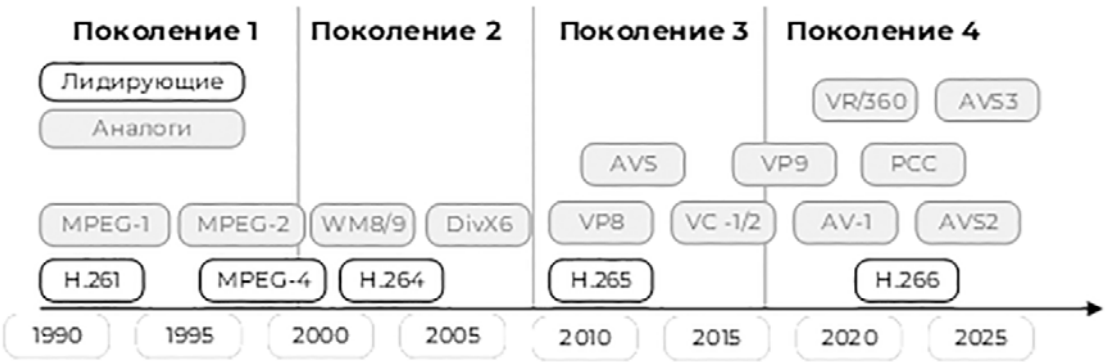


Рис. 1. Хронология появления видеокодеков

Однако в последние годы стало очевидно, что гибридные кодеки начали подбираться к пределу своих возможностей, обеспеченных традиционными приёмами. Как видно из рисунка 1, в настоящее время появляется всё больше аналогов и вариаций классических кодеков, что может свидетельство-

вать о уменьшении эффективности улучшения гибридных кодеков и поиске альтернативных подходов к решению проблемы кодирования видео. Положение могут исправить алгоритмы машинного и глубокого обучения, интегрированные в классические инструменты сжатия видеоданных.

С недавнего времени было выпущено множество различных алгоритмов видеокомпрессии, в том числе кодеки с интеграцией моделей машинного обучения.

В данной статье проведен анализ эволюции гибридной модели кодеков, а также обзор некоторых существующих алгоритмов сжатия на основе машинного обучения.

Гибридная модель кодирования видео

Модель называется гибридной, потому что построена из совокупности последовательно идущих унифицированных элементов, каждый из которых реализует свой механизм сжатия. Согласно профессору Ричардсону [6], характерными для гибридной модели инструментами являются:

- Деление изображения на макроблоки.
- Определение векторов движения.
- Компенсация движения.
- Дискретно-косинусное преобразование.
- Матрица квантования.

Начиная с H.261 до текущего момента все ведущие кодеки построены на основе гибридной модели. С тех времён алгоритмы сжатия видео не раз эволюционировали, однако большинство их улучшений носили экстенсивный характер. Данную закономерность можно явно проследить по данным, приведенным в таблице 1.

Таблица 1

Характеристики видеокодеков

Видеокодек	Количество направлений векторов движения	Точность вектора, пикселей	Количество вариаций макроблоков
MPEG-1	8	1/2	1
MPEG-2	8	1/2	2
H.264	16	1/4	3
H.265	32	1/8	4
H.266	64	1/16	5

Можно предположить, что в связи с постоянным и многократным обновлением и улучшением инструментов сжатия гибридной модели, они уже не имеют большого потенциала к развитию. Вместо этого стоит обратить внимание в другие области.

Огромный потенциал к росту эффективности кодирования лежит в областях машинного или глубокого обучения, которыми традиционные алгоритмы пренебрегают. Однако в последнее время повсеместно появляются видеокодеки, использующие различные алгоритмы из этой сферы.

AIVC: Кодирование видео на основе искусственного интеллекта

Примером решения с использованием искусственного интеллекта может послужить французский AIVC [7]. Это видеокодер, который для обучения использует модель, опубликованную в открытом доступе. AIVC способен сжимать видео с использованием любой конфигурации кодирования, состоящей из кадров I, P и B. Этот видеокодер показал себя конкурентоспособным с ведущими кодеками при определенных условиях тестирования.

Кодек включает в себя две нейронные сети, которые отвечают за компенсацию движения и кодирование соответственно. Авторы алгоритма сравнивают его с HEVC в разных конфигурациях метрикой качества MS-SSIM (рис. 2). На основе представленных графиков можно сделать несколько выводов. Во-первых – алгоритм авторов значительно превосходит традиционный HEVC при конфигурации, в которой каждый кадр является опорным. Данная конфигурация используется нечасто, так как сильно проигрывает по сжатию вариантам с межкадровым предсказанием. Во-вторых, на остальных конфигурациях AIVC показывает конкурентоспособные результаты, но только в областях высокого битрейта.

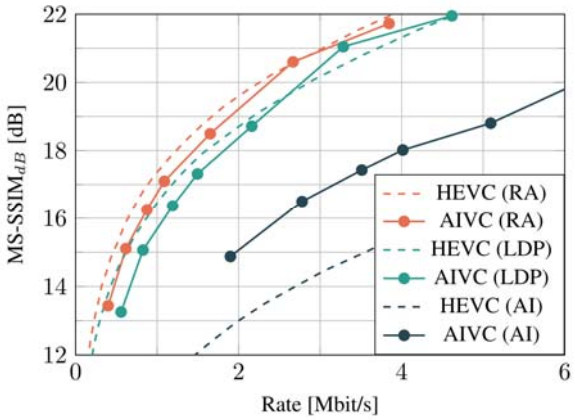


Рис. 2. Зависимость метрики качества MS-SSIM от битрейта разных конфигураций AIVC и HEVC

Упоминания стоит и дополнительный режим кодирования AIVC – режим пропуска. Он позволяет сократить количество вычислений, возложенное на нейронную сеть, ответственную за кодирование кадра. Этот режим позволяет не задействовать нейросеть кодирования в случае, если область предсказана достаточно хорошо в нейросети предсказания. Таким образом, достигается улучшение производительности AIVC.

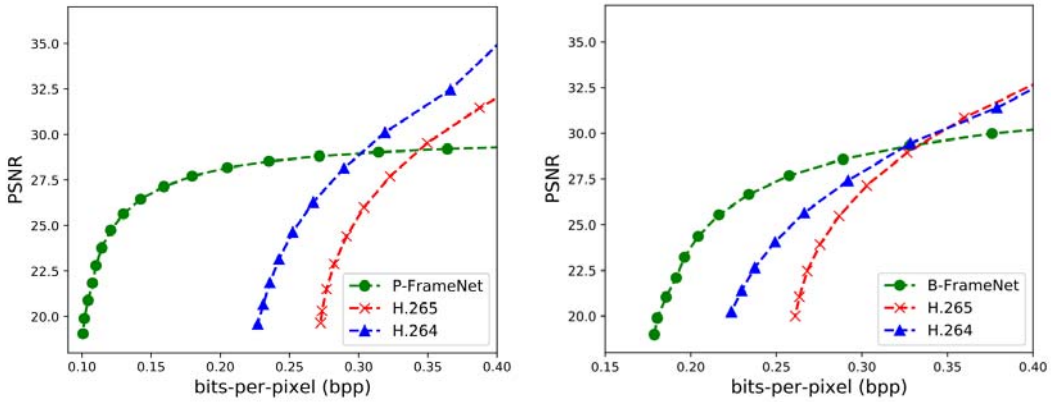
Подводя итог вышесказанному, несмотря на тот факт, что AIVC предлагает неплохую производительность, ему сложно превзойти современные классические кодеки особенно на более низких битрейтах. Авторы считают, что введение дополнительных режимов кодирования, похожих на режим пропуска улучшит результаты нейронных кодеков. Более того, представленные экспериментальные результаты подчеркивают относительную слабость компонента предсказания движения, который необходимо доработать для получения лучшей производительности.

Основанная на глубоком обучении компенсация движения параллельного межкадрового предсказания в видеокомпрессии

Один из вариантов интеграции технологий машинного обучения в видеокодеки – компенсация движения на основе глубокого обучения [8]. Авторы представили две сети глубокой оценки и компенсации движения, для B-кадров и P-кадров соответственно. Данные сети рассматриваются, как альтернатива алгоритмам блочного движения в существующих видеокодеках для улучшенного межкадрового прогнозирования.

В отличие от традиционных видеокодеков, данный алгоритм не передаёт векторы оптического потока для управления прогнозами видеокадров. Вместо этого сеть кодировщика учится идентифицировать и сжимать движение, присутствующее в видеопоследовательности напрямую. Полученный двоичный код движения используется для управления соответствующими декодерами при преобразовании содержимого опорного кадра. Это позволяет выполнять параллельную компенсацию движения, которая предсказывает более сложное движение, чем методы, основанные на потоках.

Также данный метод позволяет избавиться от артефактов блочности, появляющихся после компенсации движения, использующейся в традиционных кодеках. Благодаря этому пропадает необходимость в фильтре деблокинга, который призван сгладить блочную структуру выходного изображения.



а) б)

Рис. 3. Метрика качества PSNR P-кадров (а), B-кадров (б)

Однако данный метод пока не конкурентоспособен при приемлемом битрейте, используемом в повседневной жизни. Как видно из рисунка 3, компенсация движения на основе глубокого обучения превосходит классические решения лишь при больших коэффициентах сжатия. Это делает вышеописанный метод неприменимым в реальных условиях, ввиду невостребованности видеопотока сверхнизкого качества.

Увеличение разрешения цифровых изображений с использованием технологий машинного обучения

В последнее время стали особенно популярны инструменты, позволяющие улучшать качество изображений с помощью искусственного интеллекта. Для подобного рода преобразований используется термин «апскейл». В основе данного решения лежат методы искусственного интеллекта, которые позволяют анализировать большие объёмы информации и на основе этого делать прогнозы относительно цветов и текстур, которые должны быть включены в улучшенное изображение. Современные апскейлеры используют нейронные сети, способные анализировать мелкие детали и добавлять их в изображение, делая его более реалистичным и качественным.

Для оценки качества предсказания нейронных сетей подобного назначения, был взят набор фотографий с типичным пользовательским контентом. Далее, разрешение каждого изображения по высоте и ширине было уменьшено вдвое. Получившиеся изображения были пропущены через различные алгоритмы увеличения разрешения, чтобы вернуть фотографиям исходный формат, другими словами, восстановить. На основе восстановленных и исходных изображений была вычислена метрика качества PSNR (Пиковое отношение сигнал/ шум). В таблице 2 приведены результаты тестов нескольких современных алгоритмов увеличения разрешения.

- EDSR — Enhanced Deep Residual Networks for Single Image Super-Resolution
- MDSR — Multi-Scale Deep Super-Resolution System
- MSRN — Multi-scale Residual Network for Image Super-Resolution
- ESRGAN [9]

Таблица 2

Среднее значение PSNR для изображений, увеличенных при помощи различных алгоритмов

Видеокодек	EDSR	MDSR	MSRN	ESRGAN
$PSNR_{ср.}$	28.5	28.49	29.1	27.2

Основываясь на полученных сведениях, можно заключить, что алгоритмы, подобные описанным выше, функционируют удовлетворительно. Технология увеличения разрешения имеет большой потенциал в сжатии данных, а, следовательно, и в видеокомпрессии.

Видеокомпрессия с реставрационной обработкой на основе машинного обучения в петле обратной связи

В данной статье были освещены некоторые примеры кодеков, основанных на машинном обучении. Недостатком кодека AIVC является непостоянство конкурентоспособности с традиционными решениями. Он превосходит классический H.265 только при высокой пропускной способности. Метод компенсации движения на основе глубокого обучения, напротив, показывает превосходящие результаты только при крайне малых битрейтах, нерелевантных для обычного пользователя. Существует потребность в алгоритме, который будет показывать приемлемые результаты, не зависящие от вводных данных, таких как битрейт. Решением проблемы нестабильности может выступить классический кодек, взятый за основу и дополненный алгоритмом искусственного интеллекта.

На основании приемлемых показателей работы алгоритмов увеличения разрешения, предлагается вариант интеграции подобного алгоритма в гибридную модель – видеокомпрессия с реставрационной обработкой на основе машинного обучения в петле обратной связи.

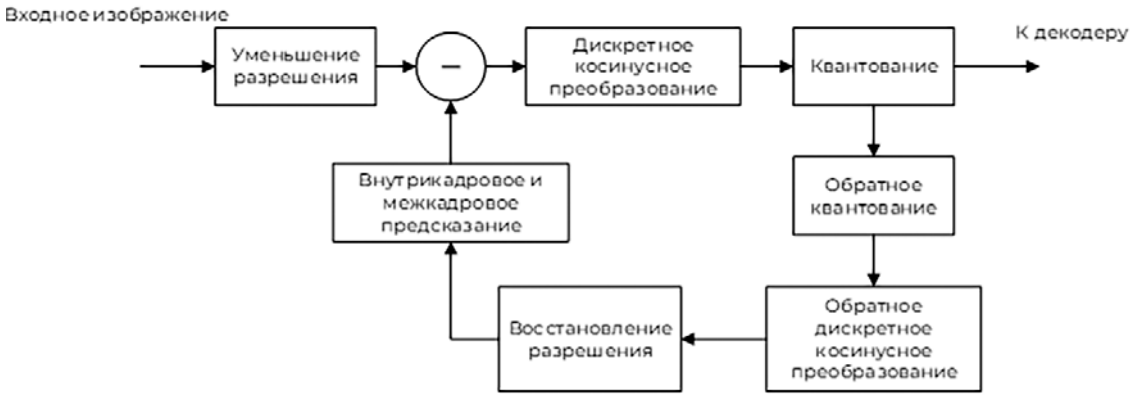


Рис. 4. Принципиальная схема предложенного кодера

Данный метод подразумевает замену фильтра деблокинга, находящегося в классических кодеках, таких как AVC и HEVC, на алгоритм улучшения качества. В свою очередь, разрешение входного изображения уменьшается вдвое, для последующей реставрации на стороне декодера. Вместе с информацией о кадре, в поток данных также будет добавлена ошибка предсказания реставрации и зерно генерации. Принципиальная схема кодера с интеграцией блока повышения разрешения представлена на рисунке 4.

При описанном выше подходе ожидаемый коэффициент сжатия посредством понижения разрешения и последующей реставрации можно вычислить по формуле:

$$K = \frac{R}{D + E}$$

K – ожидаемый коэффициент сжатия; R, D – размер исходного и сжатого изображения соответственно; E – ошибка предсказания

Такой метод не исключает применение традиционных инструментов гибридной модели видеокодеков, а добавляет блок к существующему алгоритму. Таким образом, вычисленный по формуле коэффициент сжатия измеряет эффективность только предложенного модуля. Ещё одним положительным аспектом является то, что описанный выше модуль не затронет уже существующих инструментов, обеспечивающих эталонную производительность классических кодеков.

Заключение

В данной работе был проведено исследование решений в области кодирования видео, как традиционных, так и инновационных вариантов с использованием полного или частичного машинного обучения. Освещены проблемы классического подхода, связанные с экстенсивным путем развития традиционных кодеков и малым потенциалом дальнейшего развития инструментов гибридной модели видеокодеков.

Анализ видеокодеков с имплементацией искусственного интеллекта показал, что эти решения могут быть конкурентоспособны при определенных условиях, порой превосходя HEVC. Но при качестве изображения, подходящего особенностям эксплуатации массового потребителя, кодеки проигрывают в эффективности классическим решениям.

Было проведено сравнение качества работы алгоритмов увеличения разрешения изображений. Полученные результаты показывают, что сжатие изображения и последующее его восстановление до изначального разрешения несущественные искажения по сравнению с исходным изображением. Это позволяет строить гипотезы о интеграции подобных алгоритмов в видеокодеки. Принцип одной из возможных реализаций такого кодека был представлен в статье. Сильными сторонами описанной реализации является высокий ожидаемый коэффициент сжатия и возможность интеграции в существующие решения.

Дальнейшая работа будет нацелена на программную реализацию предложенного метода, и проверку гипотезы ожидаемого коэффициента сжатия и производительности.

Литература

1. *Nezhivleva K.I., Davydova A.A., Drebuzhan A.M., Mozhaeva A.I., Balobanov A.* Comparing of Modern Methods Used to Assess the Quality of Video Sequences During Signal Streaming with and Without Human Perception // 2022 Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO), 2022, pp. 1-6, doi: 10.1109/SYNCHROINFO55067.2022.9840983.
2. ISO/IEC 13818-1*:* Information technology – Generic coding of moving pictures and associated audio information Part 1: Systems, 2022.
3. *Karwowski D., Grajek T., Klimaszewski K., Stankiewicz O., Stankowski J., Wegner K.* 20 Years of Progress in Video Compression from MPEG-1 to MPEG-H HEVC. General View on the Path of Video Coding Development // Advances in Intelligent Systems and Computing. 2017. № 525, pp. 3-15.
4. ISO/IEC FDIS 23008-2*:* Information technology – High efficiency coding and media delivery in heterogeneous environments Part 2: High efficiency video coding, 2013.
5. SO/IEC 14496-10*:* Information technology – Coding of audio-visual objects – Part 10: Advanced video coding, 2003.
6. *Iain E. Richardson*. The H.264 advanced video compression standard Iain E. Richardson. 2nd ed., 2003.
7. *Ladune, P. Philippe T*. AIVC: Artificial intelligence based video codec, 2022. DOI:10.48550/arXiv.2202.04365
8. *Andre Nortje, Herman A. Engelbrecht, Herman Kamper.* Deep motion estimation for parallel inter-frame prediction in video compression, 2019. Doi:10.48550/arXiv.1912.05193
9. *Xintao Wang, Liangbin Xie, Chao Dong, Ying Shan*. Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data, 2019. DOI:10.48550/arXiv.2107.10833
10. *Ivanchev V.V., Vlasuyk I.V., Stroganova E.P.* Objective assessment of colours' warmth // T-Comm. 2024. Т. 18. № 1. С. 44-50.
11. *Степанов Н.С., Матуа Д.Д., Мазин В.А., Вотяков С.Ю., Винецкий В.В., Власюк И.В.* Анализ текущих алгоритмов вычисления области регионов интереса пользователей при потоковой передачи видеоконтента // T-Comm: Телекоммуникации и транспорт. 2023. Т. 17. № 2. С. 27-32.
12. *Власюк И.В., Узеев А.А., Пахомова Е.А.* Исследование методов коррекции изображений с расширенным динамическим диапазоном для воспроизведения на устройствах с ограниченными параметрами отображения // Телекоммуникации и информационные технологии. 2023. Т. 10. № 1. С. 135-144.
13. *Mozhaeva A., Vashenko E., Selivanov V., Potashnikov A., Vlasuyk I., Streeter L.* Analysis of current video databases for quality assessment // T-Comm. 2022. Т. 16. № 2. С. 48-56.
14. *Власюк И.В., Киселева А.С.* Анализ эффективности безреференсных метрик применительно к оценке качества видео при потоковой передаче // Телекоммуникации и информационные технологии. 2022. Т. 9. № 2. С. 65-74.
15. *Кремлева Э.А., Власюк И.В.* Оценка эффективности методов визуализации одноканальных изображений в условных цветах // Телекоммуникации и информационные технологии. 2021. Т. 8. № 1. С. 29-37.
16. *Valitskaya N.S., Vlasyuk I.V., Potashnikov A.M.* Video compression method on the basis of discrete wavelet transform for application in video information systems with non-standard parameters // T-Comm. 2020. Т. 14. № 3. С. 47-53.
17. *Поташников А.М., Власюк И.В.* Метод построения равноконтрастного цветового пространства для заданной системы отображения информации и условий контроля // T-Comm: Телекоммуникации и транспорт. 2020. Т. 14. № 4. С. 15-22.

ИМИТАЦИОННАЯ МОДЕЛЬ АЛГОРИТМА ДИФФИ-ХЕЛЛМАНА В СОСТАВЕ КРИПТОГРАФИЧЕСКОГО ПРОТОКОЛА

Михалевич Игорь Феодосьевич
Российский университет транспорта, доцент, к.т.н., старший научный сотрудник, Москва, Россия
mif-orel@mail.ru

Потапов Артем Константинович
Российский университет транспорта, студент, Москва, Россия
artem_potapov_1975@mail.ru

Соколов Илья Дмитриевич
Российский университет транспорта, студент, Москва, Россия
filogssss@gmail.com

Ковров Артем Игоревич
Российский университет транспорта, студент, Москва, Россия
kovorv_tema@mail.ru

Аннотация
В статье рассматривается имитационная модель алгоритма Диффи-Хеллмана в составе криптографического протокола, содержащего реализацию симметричного шифра, процедуры проверки чисел на простоту и поиска первообразных корней простого модуля. Показано, что успешное применение криптографических методов требует знаний в математике, теории чисел и криптографии. Представлена программная реализация на Python с PySide6, визуализирующая пошаговые вычисления, что делает имитационную модель полезной для обучения основам криптографии и применения в исследовательских целях.

Ключевые слова
Алгоритм обмена ключами, Диффи-Хеллман, криптография, первообразный корень, проверка простоты, симметричное шифрование, шифр Сцитало

Введение

Современная криптография опирается на комбинированное использование асимметричных алгоритмов для обмена ключами и эффективных симметричных шифров для защиты данных. В их основе лежат сложные математические модели, пониманию которых способствуем имитационное моделирование [1].

Алгоритм Диффи-Хеллмана (DH) [2] позволяет двум сторонам согласовать общий секретный ключ, используя открытый канал, что является одним из ключевых элементов различных криптографических систем Он широко применяется в протоколах HTTPS (TLS/SSL), SSH, VPN и в гибридных системах шифрования, где сгенерированный общий секрет затем применяется в симметричных алгоритмах, таких как AES [3], Магма, Кузнечик [4] др.

При этом использование криптографических средств в Российской Федерации регулируется рядом нормативных правовых актов и стандартов. Например, защита персональных данных в соответствии с Федеральным законом №152-ФЗ «О персональных данных» предполагает применение сертифицированных средств криптографической защиты. Таким образом, внедрение и применение алгоритма Диффи-Хеллмана в реальных системах может потребовать соответствия отечественным стандартам и прохождения процедур оценки соответствия. Для успешной работы алгоритма DH необходимы большие простые числа и первообразные корни по модулю этого числа [5, 6]. После выработки ключа можно использовать симметричный шифр для передачи сообщений. В качестве примера в данной работе рассматривается шифр Сцитало [6, 7], демонстрирующий основы перестановочного шифрования.

Целью работы является разработка имитационной модели (ИМ), позволяющей визуализировать процедуры проверки чисел на простоту, поиска первообразных корней, выработки ключей по DH и их использования для симметричного шифрования. Разработанная ИМ будет полезна в учебных и исследовательских целях, облегчая понимание сложных теоретических основ криптографии.

Выбор моделируемой системы

В качестве объекта моделирования выбран алгоритм Диффи-Хеллмана для выработки общего секретного ключа, поскольку он широко применяется на практике в криптографических системах. Для шифрования сообщений с использованием полученного ключа применяется шифр Сцитало [6, 8], который преобразует исходный текст на основе перестановок. Дополнительной задачей, решаемой в процессе моделирования, является поиск первообразных корней для числа p, используемых при генерации ключей [5, 6].

Теоретические основы

2.1. Алгоритм Диффи-Хеллмана

Алгоритм DH позволяет двум участникам, обычно именуемым «Алиса» и «Боб», договориться о секретном ключе, используя открытый канал [2]. Основными этапами являются:

1. Алиса выбирает большое случайное число a, а Боб – число b.
2. Стороны договариваются о простом числе p и его первообразном корне g.
3. Алиса вычисляет $A=g^a \bmod p$, и передает A Бобу; Боб вычисляет $B=g^b \bmod p$ и передает B Алисе.
4. Получив ключ другой стороны, Алиса вычисляет $K=B^a \bmod p$, а Боб вычисляет $K=A^b \bmod p$. Ключи совпадают из-за свойства коммутативности возведения в степень по модулю: $(g^a)^b=(g^b)^a=g^{ab} \bmod p$.

Общая схема системы конфиденциального обмена сообщениями с использованием алгоритма Диффи-Хеллмана для выработки общего ключа приведена на рисунке 1.

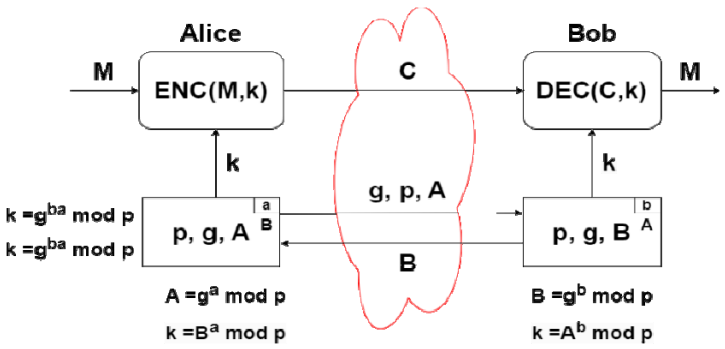


Рис. 1. Общая схема системы конфиденциального обмена сообщениями с DH-выработкой ключа

В данной системе Алиса и Боб согласуют общий ключ k, который используется для шифрования и дешифрования сообщений. Параметры g, p и открытые ключи A, B передаются через открытую линию связи. Алиса и Боб вычисляют общий ключ k с помощью своих секретных ключей a и b, используя свойства модульной арифметики. Далее сообщение M шифруется функцией ENC (M, k) на стороне Алисы и передается Бобу в зашифрованном виде C. Боб, получив сообщение, расшифровывает его с использованием DEC (C, k), восстанавливая исходное сообщение M. Такой подход обеспечивает безопасность передачи информации, так как общий ключ k никогда не передается напрямую и вычисляется только на стороне участников.

На рисунке 1 применены следующие обозначения:

a, B, b, B – закрытые и открытые ключи Алисы и Боба соответственно;

p – простое число;

g – первообразный корень кольца вычетов по модулю p;

k – общий секретный ключ Алисы и Боба;
M – сообщение;
C – зашифрованное сообщение;
ENC – функция шифрования сообщения;
DEC – функция расшифровки сообщения;
DH – алгоритм Диффи-Хеллмана.

2.2. Шифр Сцитало

Шифр Сцитало – простой перестановочный метод шифрования, известный со времен Древней Спарты [6]. Идея состоит в наматывании ленты с текстом на цилиндр определенной толщины и последующем чтении текста по вертикали. В предложенном примере шифр Сцитало реализуется как преобразование исходного сообщения в двумерную матрицу символов и считывание результата по другому направлению, используя общий ключ как параметр перестановки.

2.3. Первообразные корни

Первообразный корень мультипликативной группы кольца вычетов числа p – это такой элемент g, что порождает все числа мультипликативной группы чисел по модулю p. Для алгоритма Диффи-Хеллмана выбор первообразного корня g по модулю p важен для генерации ключей.

2.4. Наивный тест проверки простоты

Наивный тест простоты является самым простым и интуитивно понятным методом. Его суть заключается в проверке числа n на делимость всеми числами от 2 до $\sqrt{n}$. Если n делится хотя бы на одно число без остатка, то оно составное; если нет – простое.

Преимущества – простота реализации, подходит для небольших чисел. Недостатки – экстремально медленный для больших чисел, так как выполняет большое количество операций деления.

Применимость – используется для обучающих целей или в ситуациях, где требуется проверить небольшие числа.

2.5. Тест Ферма

Тест Ферма основан на малой теореме Ферма, которая утверждает: если n – простое число и a – любое число, взаимно простое с n, то $a^{n-1} \equiv 1 \mod n$. Тест проверяет, выполняется ли это свойство для нескольких значений a.

Алгоритм:

1. Выбрать случайное a из диапазона $[2, n-2]$.
2. Вычислить $a^{n-1} \mod n$.
3. Если результат не равен 1, n составное.

Проблема – существуют составные числа, называемые числами Кармайкла, которые удовлетворяют тесту Ферма для любого a, взаимно простого с n.

Применимость – простой и быстрый тест, но недостаточно точный. Используется в комбинации с другими тестами.

2.6 Тест Штрассена

Тест Штрассена основан на идеях из линейной алгебры и арифметики. Он проверяет свойства чисел через сложные математические операции, такие как разложение числа и проверка его поведения в специфических условиях.

Особенности – этот тест менее популярен из-за сложности реализации по сравнению с другими алгоритмами, но он обладает хорошими свойствами для некоторых специфических чисел.

Применимость – применяется редко, чаще в учебных и исследовательских целях.

2.7 Тест Миллера-Раббина

Тест Миллера-Раббина – один из самых популярных вероятностных тестов простоты [4, 5]. Он основывается на разложении числа $n-1$ в виде $2^s \cdot d$, где d нечетное. Затем тест проверяет определенные свойства возведения в степень по модулю, которые характерны для простых чисел.

Алгоритм:

1. Представить $n-1$ как $2^s \cdot d$, где d нечетное.
2. Выбрать случайное a из диапазона $[2, n-2]$.
3. Проверить $a^d \bmod n$ и значения $a^{2^r d} \bmod n$ для $r=0,1,\ldots, s-1$

Преимущества – высокая точность при многократном повторении (вероятность ошибки уменьшается экспоненциально с количеством итераций). Быстрота и простота реализации для больших чисел.

Применимость – широко используется на практике, в том числе в криптографии для проверки больших чисел.

Описание имитационной модели алгоритма Диффи-Хеллмана

Разработанная ИМ выполнена в виде кроссплатформенного приложения, реализованного на языке Python с использованием библиотеки PySide6 [9] для создания графического интерфейса [10, 11]. Благодаря наглядной визуализации всех процедур пользователь может поэтапно наблюдать за ходом вычислений, связанных с проверкой простоты чисел (тест Миллера-Рабина), поиском первообразных корней, реализацией алгоритма Диффи-Хеллмана, а также за работой перестановочного шифра Сцитало. Такой подход значительно упрощает понимание сложного математического аппарата, лежащего в основе современных криптографических систем.

В процессе реализации были разработаны отдельные классы и функции, каждый из которых отвечает за свою часть вычислительного процесса. Структура кода организована таким образом, чтобы выделить следующие основные компоненты:

- PrimitiveRootFinder – класс для поиска первообразных корней по модулю простого числа.
- DiffieHellman – класс, отвечающий за реализацию алгоритма Диффи-Хеллмана: генерацию открытых и закрытых ключей, а также вычисление общего секретного ключа.
- ScytaleCipher – класс, реализующий перестановочное шифрование текста с использованием общего ключа, сгенерированного по алгоритму Диффи-Хеллмана.

Для графического интерфейса применяются средства PySide6 [9; 10], позволяющие создавать настраиваемые окна, кнопки, поля ввода и вывода, таблицы и другие элементы. Данный инструментарий обеспечивает независимость внешнего вида приложения от операционной системы и дает возможность гибкой адаптации интерфейса под нужды пользователя. В отличие от традиционных библиотек, использующих нативный код, PySide6 обеспечивает кроссплатформенную совместимость за счет прямой интеграции с фреймворком Qt и позволяет динамически изменять внешний вид приложения.

Сочетание приведенных компонентов программного кода и удобного графического интерфейса позволило в полной мере добиться поставленной цели – обеспечить пошаговую визуализацию вычислительных процедур, применяемых в рамках алгоритма Диффи-Хеллмана, проверки простоты, выбора первообразных корней и шифрования шифром Сцитало. Каждому этапу соответствует отдельная функциональность интерфейса, благодаря чему пользователь может не только наблюдать, но и экспериментировать с параметрами входных данных, лучше осознавая внутреннюю логику работы криптосистемы.

Ошибки и трудности при выборе параметров алгоритма Диффи-Хеллмана

Рассмотрим ошибки и трудности, возникающие при выборе параметров алгоритма DH.

Неправильный выбор простого числа p.

Алгоритм Диффи-Хеллмана требует большого простого числа p. Например, RSA был впервые описан в 1977 году как первый широко известный практичный алгоритм шифрования с открытым ключом. На заре своего существования RSA использовал небольшие размеры ключей, около 67-68 символов (примерно 512 бит). Это объяснялось ограничениями вычислительных мощностей тех лет. Такие ключи обеспечивали достаточную защиту для многих задач, поскольку атаки, основанные на факторизации больших чисел, были крайне сложны для выполнения на доступном оборудовании. Однако со временем, с развитием технологий и увеличением вычислительных мощностей, ключи такого размера стали небезопасными. Современные стандарты требуют ключей размером от 2048 бит и выше для обеспечения стойкости к современным методам криптоанализа.

В учебных целях студенты берут числа небольшого размера или, что хуже, составные числа. Это приводит к тому, что алгоритм перестает быть безопасным, так как атаки (например, методом грубой силы) становятся тривиальными.

Выбор g – первообразного корня.

При неправильном выборе g (например, не являющегося первообразным корнем мультипликативной группы вычетов кольца вычетов) возникают проблемы с генерацией всех элементов множества вычетов. Это ограничивает ключи, которые могут быть сгенерированы, и делает алгоритм уязвимым.

Малые значения a и b.

Если секретные ключи a и b слишком малы, алгоритм становится подверженным атакам перебором. Студенты иногда выбирают a=2 или a=3, что нарушает условия криптографической стойкости.

Пренебрежение случайностью.

Студенты могут выбирать a и b вручную, без использования генераторов случайных чисел, что снижает энтропию и безопасность алгоритма.

Цикличность классов в кольцах вычетов.

Кольцо вычетов по модулю p (где p — простое число) формирует мультипликативную группу порядка $p-1$. Цикличность этой группы означает, что любой элемент g, являющийся первообразным корнем, может генерировать все остальные элементы группы через возведение в степени.

Ошибочное понимание цикла.

Студенты могут ошибочно предполагать, что любое a, взаимно простое с p, является g. Однако, если a не является первообразным корнем, результат возведения в степени может покрывать лишь подгруппу множества вычетов. Это приводит к невозможности корректной генерации всех ключей алгоритмом Диффи-Хеллмана.

Практический пример. Для p=7 первообразным корнем g являются числа 3 и 5. Если принять за g число 2, то $2^k \bmod 7$ даст значения 1, 2, 4. При этом не будут созданы все элементы группы и возникнет цикличность созданных элементов.

Свойства элементов мультипликативной группы кольца вычетов по модулю p=7, демонстрирует рисунок 2, на котором видна цикличность элементов, не являющихся примитивными корнями) и принадлежность только определенных элементов к примитивным корням. Такие элементы удобно называть также генераторами группы.

	Значения степеней числа по модулю p							
Элемент	1	2	3	4	5	6	Порядок группы	Первообразный корень?
1	1	1	1	1	1	1	1	Нет
2	2	4	1	2	4	1	3	Нет
3	3	2	6	4	5	1	6	Да
4	4	2	1	4	2	1	3	Нет
5	5	4	6	2	3	1	6	Да
6	6	1	6	1	6	1	2	Нет

Рис. 2. Демонстрация поиска генераторов группы при p=7

Поясним рисунок 2.

1. Столбцы 1–6. В них представлены остатки от возведения чисел (элементов) в степени k и последующего вычисления остатка по модулю 7.

Остатки вычисляются по общей формуле: $n^k \bmod m$, где:

- n – элемент таблицы;
- k – степень (от 1 до p-1, где p = 7);
- m = 7 – модуль.

2. Признак первообразного корня.

Если остатки для элемента перебирают все числа от 1 до p-1 (в данном случае от 1 до 6), то этот элемент является первообразным корнем.

Такой элемент выделен в таблице, и в последнем столбце указано "Да".

3. Общее количество уникальных значений.

В предпоследнем столбце указано количество уникальных остатков, полученных для каждого элемента при возведении его в степени.

4. Последний столбец показывает, является ли элемент первообразным корнем. Если все числа от 1 до p-1 появляются в остатках, то в столбце стоит "Да", в противном случае – "Нет".

5. Цветовая маркировка.

Красным выделены повторяющиеся значения остатков, которые показывают, что элемент не является первообразным корнем.

Зеленым выделяются все уникальные значения, которые подтверждают, что элемент является первообразным корнем.

6. Цель таблицы.

Проверить, какие элементы являются первообразными корнями по модулю 7, а также проанализировать распределение остатков.

Графический интерфейс и функционирование ИМ

При запуске ИМ выполняется следующая последовательность действий:

1. Ввод исходных данных: сообщение, простое число p.
2. Поиск и выбор первообразного корня g, используемого для генерации ключей.
3. Генерация ключей по алгоритму Диффи-Хеллмана: расчет открытых ключей A и B, обмен параметрами и вычисление общего ключа K.
4. Шифрование Сцитало: исходное сообщение преобразуется на основе полученного ключа.

Промежуточные результаты отображаются в текстовом и графическом виде, что позволяет пользователю последовательно анализировать каждый этап выполнения алгоритмов.

Графический интерфейс на PySide6 обеспечивает удобную визуализацию всех этапов [9; 10]. Пользователь может ввести исходные данные (сообщение, числовой параметр для генерации p), пошагово следить за процессом проверки простоты, выбора первообразного корня, генерации ключей и шифрования. Для наглядности в интерфейсе используются таблицы, кнопки управления шагами, текстовые поля вывода промежуточных результатов и графические элементы.

Работа алгоритма Диффи-Хеллмана показана на рисунке 3.

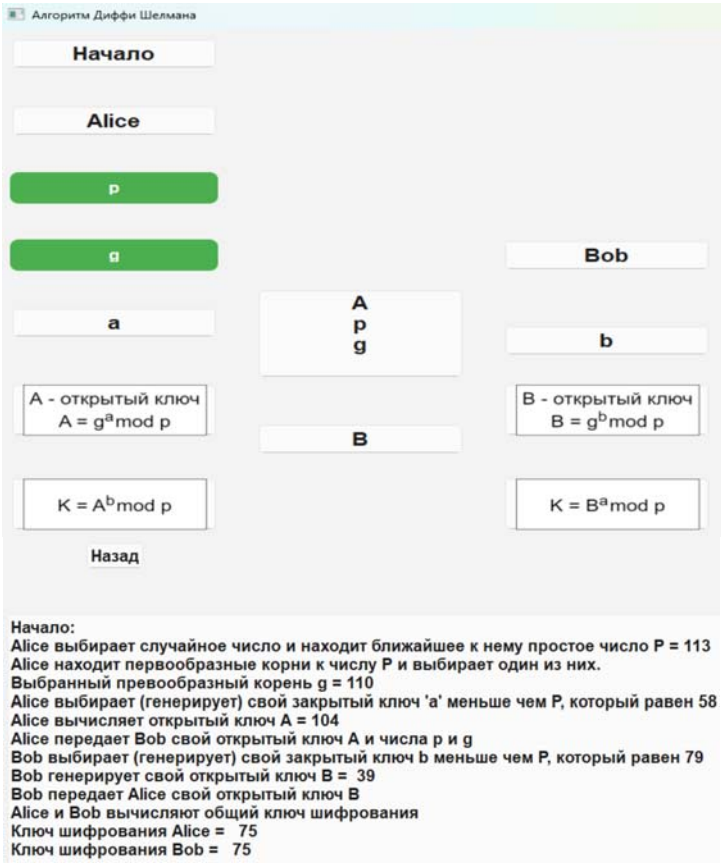


Рис. 3. Работа ИМ по алгоритму Диффи-Хеллмана

Интерфейс наглядно отображает выбор числа p (простое число), примитивного корня g, а также закрытых ключей a и b, используемых для генерации открытых ключей A и B. Далее происходит обмен открытыми ключами и вычисление общего ключа шифрования K обеими сторонами. Такой пошаговый подход с текстовым выводом и интерактивными элементами позволяет детально проследить весь процесс и понять основные принципы алгоритма Диффи-Хеллмана.

Процесс шифрования сообщения с использованием метода «Сцитало» показан на рисунке 4. Визуализация включает изображение цилиндра с направлением записи сообщения, который символизирует шифровальную машину, а также таблицу, представляющую зашифрованное сообщение в виде упорядоченной матрицы. Показан ключ шифрования (например, диаметр цилиндра), используемый для определения длины строки. Программа наглядно отображает, как исходное сообщение преобразуется в зашифрованный текст, следуя принципам метода. Данный подход позволяет понять механизм шифрования и на практике наблюдать его реализацию, что важно для демонстрации методов классической криптографии.

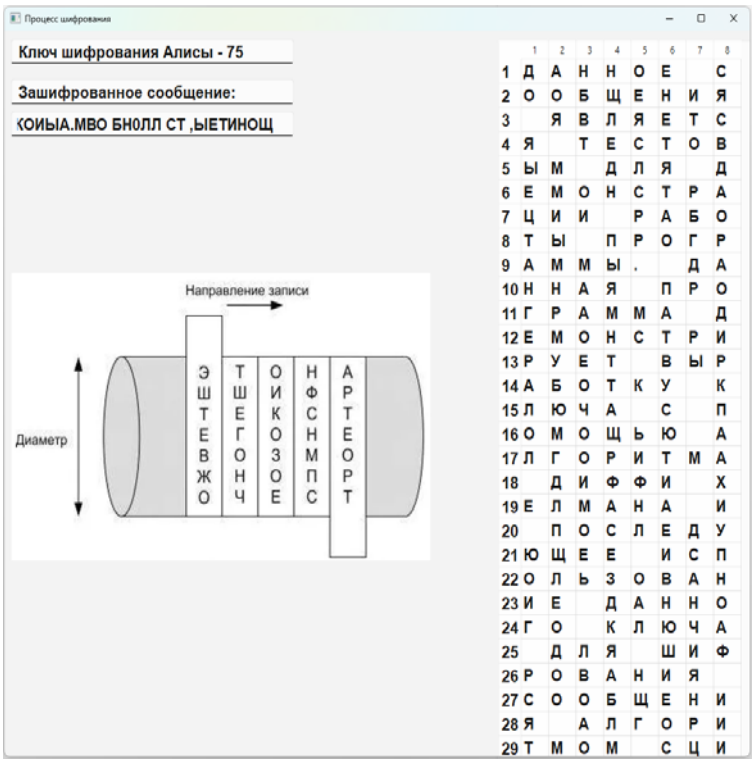


Рис. 4. Шифрование методом Сцитало

Решения, применимые для моделирования других криптографических систем

Принятые архитектурные решения могут быть легко адаптированы для моделирования других криптографических алгоритмов. Использование гибкой структуры и визуального интерфейса позволяет демонстрировать работу не только алгоритма Диффи-Хеллмана, но и более сложных систем, таких как RSA или AES. Данный подход наглядно иллюстрирует внутренние процессы криптосистем и упрощает обучение пользователей на всех уровнях подготовки.

Заключение

Разработанная имитационная модель демонстрирует процесс выработки общего секретного ключа по алгоритму Диффи-Хеллмана, начиная от проверки простоты чисел до использования ключа для симметричного шифрования. Каждый этап подробно визуализирован, что позволяет пользователям (студентам, исследователям) лучше понять, как теоретические концепции реализуются на практике.

Особое внимание уделено интеграции нескольких криптографических методов:

- поиск первообразных корней для заданного простого числа, что является ключевым этапом для корректного функционирования алгоритма Диффи-Хеллмана;
- реализация самого алгоритма Диффи-Хеллмана, включая генерацию открытых и закрытых ключей, обмен данными между сторонами и вычисление общего ключа;
- шифрование шифром Сцитало, который, будучи историческим примером перестановочного шифра, демонстрирует основы симметричных криптосистем.

Благодаря использованию графического интерфейса, созданного на основе PySide6, пользователи могут не только наблюдать, но и активно взаимодействовать с моделью, экспериментируя с параметрами, такими как выбор простых чисел, первообразных корней и ключей. Такой подход значительно облегчает восприятие сложных математических концепций и способствует лучшему пониманию внутренней логики работы криптографических алгоритмов.

Имитационная модель имеет высокую образовательную ценность. Она помогает студентам избежать типичных ошибок, таких как:

- неправильный выбор простых чисел и первообразных корней;
- понимание ограничений методов проверки простоты;
- осознание важности случайности при генерации ключей.

Кроме того, модель может быть использована в исследовательских целях для изучения новых криптографических подходов, их безопасности и эффективности.

Таким образом, предложенная модель не только упрощает изучение криптографии, но и открывает возможности для ее дальнейшего применения в реальных задачах, включая обеспечение безопасности данных в современных информационных системах. Она может быть легко адаптирована для моделирования других криптографических алгоритмов, таких как RSA, ECC и AES, что делает ее универсальным инструментом для обучения и исследований.

Литература

1. *Михалевич И.Ф., Абызов А.А., Архипов А.М., Басюк С.А.* и др. Имитационная модель SP-сети // REDS: Телекоммуникационные устройства и системы. 2023. № 2. С. 4-12.

2. *Diffie W., Hellman M*. New Directions in Cryptography // IEEE Transactions on Information Theory. 1976. 22(6).

3. *Daemen J., Rijmen V.* AES Proposal: Rijndael. Note on naming. https://csrc.nist.gov/csrc/media/projects/cryptographic-standards-and guidelines/documents/aes-development/rijndael-ammended.pdf#page=1 (доступ 21.12.2024).

4. ГОСТ Р 34.12-2015. Информационная технология. Криптографическая защита информации. Блочные шифры.

5. *Miller G.* Riemann’s Hypothesis and Tests for Primality // Journal of Computer and System Sciences. 1976.

6. *Rabin M*. Probabilistic Algorithm for Testing Primality // Journal of Number Theory. 1980.

7. *Bogdanov A., Khovratovich D., Rechberger Ch*. Biclique Cryptanalysis of the Full AES. https://www.researchgate.net/publication/221326929_Biclique_Cryptanalysis_of_the_Full_AES (доступ 24.12.2024).

8. *Stallings W*. Cryptography and Network Security: Principles and Practice. 5th ed. Pearson. – 2011.

9. Документация по PySide6: https://doc.qt.io/qtforpython/ (доступ 03.10.2024).

10. Java Swing – Пример реализации графического интерфейса: https://docs.oracle.com/javase/tutorial/uiswing/ (доступ 03.10.2024).

11. Регулярные выражения JAVA. Сайт о программировании https://metanit.com/java/tutorial/7.4.php (доступ 03.10.2024).