

DOI 10.37882/2223-2966.2025.10.04

КЛАССИФИКАТОР ТЕКСТА С ИСПОЛЬЗОВАНИЕМ ГИБРИДНОГО ПОДХОДА: КЛАССИЧЕСКОЙ ПРЕДОБРАБОТКИ ДАННЫХ И КВАНТОВЫХ МОДЕЛЕЙ SVM

TEXT CLASSIFIER USING A HYBRID APPROACH: CLASSICAL DATA PREPROCESSING AND QUANTUM SVM MODELS

**N. Babasanova
A. Kanev
V. Mikheeva**

Summary. The work addresses the problem of the lack of practical implementations of quantum algorithms for natural language processing (NLP). The goal of the work is to present an approach to building a quantum text classifier for the binary classification of SMS messages with a split in computations: data preprocessing is performed on a classical computer, while the classification stage is carried out using a quantum algorithm. A comparison with a classical SVM (scikit-learn) was conducted for the quantum PegasosQSVC and QSVC algorithms; with identical data preprocessing, the quantum algorithms demonstrate comparable accuracy: 94 % for the classical SVC and 96 % for PegasosQSVC (on the full dataset), and 97 % for SVC and 96 % for QSVC (on a reduced dataset).

Keywords: machine learning, NLP, quantum computing, SVM, quantum machine learning, Qiskit.

Бабасанова Надежда Сергеевна

Московский государственный технический
университет им. Н.Э. Баумана
bns22k060@student.bmstu.ru

Канев Антон Игоревич

Старший Преподаватель, Московский государственный
технический университет им. Н.Э. Баумана
aikanev@bmstu.ru

Михеева Валерия Алексеевна

Московский государственный технический
университет им. Н.Э. Баумана
mikheevava@bmstu.ru

Аннотация. Статья посвящена проблеме недостатка практических реализаций квантовых алгоритмов для обработки естественного языка (NLP). Целью работы является представление подхода к построению квантового текстового классификатора для задачи бинарной классификации SMS-сообщений с разделением вычислений: предобработка данных происходит на обычном компьютере, этап классификации происходит с использованием квантового алгоритма. Проведено сравнение с классическим SVM (scikit-learn) квантовых алгоритмов PegasosQSVC и QSVC; при идентичной предобработке данных квантовые алгоритмы демонстрируют сопоставимую точность: 94 % для классического SVC и 96 % PegasosQSVC (при полном датасете) и 97 % для SVC и 96 % QSVC (при уменьшенном датасете).

Ключевые слова: машинное обучение, NLP, квантовые вычисления, SVM, квантовое машинное обучение, Qiskit.

Введение

Современные подходы к развитию вычислительной техники сталкиваются с серьезными ограничениями, обусловленными исчерпанием ресурсов традиционной кремниевой технологии. Закон Мура, согласно которому плотность транзисторов на кристалле удваивается каждые 18–24 месяцев, утратил свою актуальность, поскольку дальнейшее уменьшение размеров транзисторов ограничивается физически и технологически. Вследствие этого внимание исследователей и ведущих технологических компаний всё чаще обращается к альтернативам, в частности — к квантовым вычислениям.

Устойчивый прогресс в области нанотехнологий и производственных процессов, а также высокая теоретическая эффективность квантовых алгоритмов в задачах линейной алгебры, особенно при работе

с пространствами больших размерностей, и в ряде узкоспециализированных задач, в частности, при генерации сложных выборок, открывает перспективы для прорыва в областях, традиционно требующих ресурсных затрат с экспоненциальной сложностью.

Хотя создание универсальных квантовых компьютеров остается долгосрочной целью, устройства промежуточного масштаба (NISQ) уже сегодня предоставляют экспериментальные платформы для отработки гибридных подходов. Характеризуясь ограниченным числом кубитов (до десятков или сотен, например, бэкэнд IBM Brisbane предоставляет 127 кубитов) и подверженностью шумам, что приводит к ограничениям размеров квантовых цепей, они уже демонстрируют вычислительные преимущества в квантово-классических алгоритмах. Учитывая интенсивное развитие данной области, можно ожидать, что в будущем появятся более стабильные и эффективные реализации NISQ-компьютеров.

Современные исследования демонстрируют эффективность различных методов машинного обучения для задач NLP, например, для задачи анализа тональности. В статье [1] проведен сравнительный анализ нескольких подходов, включая логистическую регрессию, метод опорных векторов (SVM), случайный лес и рекуррентные нейронные сети. Результаты показали, что лучшие показатели f1-меры достигаются при использовании логистической регрессии и стохастический градиентный спуск (SGD).

Однако классические методы сталкиваются с существенными ограничениями при работе с высокоразмерными данными, особенно метод опорных векторов. Как показано в работе [1], SVM является одним из самых длительных в обучении методов, поскольку плохо масштабируется с увеличением объема данных. Это вполне объяснимо тем, что этап обучения SVM имеет сложность по времени работы $O(n^3)$ и по объему дополнительной памяти $O(n^2)$, n — количество объектов [2].

Для преодоления ограничений классического SVM, связанных с вычислительными сложностями, предлагается использовать гибридные квантово-классические алгоритмы. В условиях современных NISQ-устройств это требует разработки специализированного конвейера обработки данных. Примером такого конвейера является модель Прабхат Санти, Камакхья Мишра, Сибабрата Моханти. В своей работе «Квантовый текстовый классификатор» авторы описывают и предлагают гибридный конвейер [3]. Так, они производят классическую предобработку данных: очистку текста, преобразование в векторы и снижение размерности; квантовое кодирование признаков и применение моделей классификации Qiskit (с использованием 2 кубитов): квантовый SVM (QSVC), вариационный квантовый классификатор (VQC), квантовый нейросетевой классификатор (QNNC). Все три модели выполнялись на симуляторе QASM. В результате QSVC достигает 89 % точности (accuracy), 89 % точности (precision), 89 % полноты покрытия (recall), 89 % f1-меры (против 90 % у классического SVC по тем же метрикам), VQC — 81 % по метрикам, указанным выше, QNNC уступает предыдущим алгоритмам (57 % — accuracy, 70 % — precision, 57 % — recall, 53 % — f1-мера). Авторы делают вывод, что гибридный подход уже близок к классическим методам и имеет потенциал при расширении числа кубитов, улучшении ранжирования признаков и настройки анзаца.

Цель данной работы — создание гибридного подхода: классической предобработки данных и квантового текстового классификатора с использованием большего числа кубитов для повышения точности квантовых моделей.

Решаемые задачи включают:

1. Экспериментальный подбор размерности признакового пространства, балансирующий между информативностью и требованиями NISQ-ограничений.
2. Сравнительную оценку по метрикам и времени выполнения предложенного решения с использованием библиотеки Qiskit против классического подхода, предложенного scikit-learn.

Материалы, методы решения задачи и принятые допущения

Преобразование текста в векторное представление

BERT (Bidirectional Encoder Representations from Transformers) — модель для задач обработки естественного языка.

BERT использует трансформер (механизм «внимания», который изучает контекстуальное отношение слов в тексте). Стандартный трансформер состоит из кодирующей и декодирующей части. Кодировщик считывает введенный текст, декодировщик выдает прогноз для задачи. BERT использует только кодирующую часть. Ключевым отличием данной модели является двунаправленное моделирование контекста, реализованное через задачу Masked Language Modeling и Next Sentence Prediction для двунаправленного анализа контекста [4], так BERT способен работать с многоязычными данными [5].

Модификацией BERT является Sentence-BERT (SBERT) — модель для получения векторного представления предложений. Модель использует сиамскую архитектуру с общими весами BERT-кодировщика. SBERT применяет mean-pooling к выходным эмбедингам токенов последнего слоя BERT, преобразуя их в вектор фиксированной размерности (768 или 384; в данной работе используется модель с 384 выходными векторами). SBERT дообучается на задачах Natural Language Inference, что позволяет получать семантически значимые эмбединги с линейной сложностью поиска $O(n)$ и возможностью предварительного индексирования векторов [6].

Снижение размерности

После BERT пространство признаков расширяется до пространства размерности 384, чтобы избежать «проклятия размерности» и большого потребления ресурсов, снизим размерность. Для этого воспользуемся UMAP, так как алгоритм сохраняет глобальную структуру, что может быть важно при проецировании пространства большой размерности на пространство меньшей размерности, а также по причине использования алгоритма, сильно зависящего от расстояний между объектами.

UMAP основан на трех принципах: равномерное распределение данных на римановом многообразии,

постоянство локальной метрики в малых окрестностях и локальная связность многообразия. Алгоритм строит взвешенный граф на основе k -ближайших соседей ($k=15$) с адаптивным вычислением расстояний для учета локальной плотности данных. Веса ребер определяются по формуле (1) [7].

$$w((x_i, x_{ij})) = \exp\left(\frac{-\max(0, d(x_i, x_{ij}) - \rho_i)}{\sigma_i}\right) \quad (1)$$

где ρ_i — расстояние до ближайшего соседа,
 σ_i — решение уравнения (2)

$$\sum_{j=1}^k \exp\left(\frac{-\max(0, d(x_i, x_{ij}) - \rho_i)}{\sigma_i}\right) = \log_2 k \quad (2)$$

где k — число ближайших соседей

Оптимизация проводится минимизацией кросс-энтропии между нечеткими множествами исходного и целевого графов с использованием стохастического градиентного спуска. Начальные позиции инициализируются спектральными методами.

Выбор UMAP обусловлен способностью сохранять глобальную структуру данных при работе с нелинейными зависимостями, что критично для алгоритмов, основанных на метриках расстояний.

Для представления пространства после Sentence-BERT в более компактном виде размерность данных снижена до 12 компонентов. Это число найдено экспериментальным путем, так как использование большего количества компонентов приводит к исчерпанию ОЗУ, доступного пользователю в Google Colab. В [8] упоминается, что при симуляции квантовых систем для n компонентов требуется хранить c^n комплексных чисел, c — количество состояний системы, для кубитов $c = 2$. Сложность вычислений экспоненциально растет с количеством кубитов в «наивном» случае. Также для одного из квантовых алгоритмов, а именно QSVC, необходимо вычислять квантовое ядро, для которого рассчитывается точность воспроизведения одного состояния другим для каждой пары объектов из выборки [9].

SVM

Наиболее распространенными QML фреймворками являются квантовые ядерные методы (Quantum kernel methods) и вариационный квантовый алгоритм (Quantum variational algorithm, QVA). Рассмотрим в данной работе квантовый ядерный метод, а именно квантовый метод опорных векторов, так как его можно сравнить с классической реализацией SVM в scikit-learn.

Ядерные методы — алгоритмы, использующие функцию ядра $K: X \times X \rightarrow \mathbb{R}$, которая симметрична и неотрица-

тельно определена. Общая формула ядра представлена в формуле 3.

$$K(x_i, x_{ij}) = \langle \varphi(x_i), \varphi(x_j) \rangle \quad (3)$$

где: $\varphi(x)$ — фиксированное преобразование пространства признаков.

Рассмотрим SVM в классическом понимании для задачи бинарной классификации в общем случае. Он основан на поиске оптимальной разделяющей гиперплоскости в пространстве признаков. Линейная модель SVC будет иметь вид (формула 4).

$$y(x) = \text{sign}\left(\sum_{i=1}^n \alpha_i y_i K(x, x_i) + b\right) \quad (4)$$

где: $K(x, x_i)$ — ядро, вычисляемое по (3).

Для линейных случаев в (3) и (4) $\varphi(x)$ — тождественное отображение.

Для нелинейных случаев считаем $\varphi(x)$ в (3) и (4) как $\varphi(x): \mathbb{R}^d \rightarrow H$, где H — пространство признаков большой размерности. Будем считать данное пространство Гильбертовым, так как на нем определено скалярное произведение.

Для обучения SVC решается двойственная задача оптимизации в общем случае (формула 5).

$$\begin{cases} L = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \rightarrow \max_{\alpha}; \\ 0 \leq \alpha_i \leq C, i = 1, \dots, n; \\ \sum_{i=1}^n \alpha_i y_i = 0. \end{cases} \quad (5)$$

Квантовый SVM

В работе рассматриваются чистые состояния кубитов и систем, так как рассматриваются симуляторы NISQ-устройств. В реальном NISQ-устройстве имеется шум, который влияет на состояния кубитов и результаты.

Эволюция квантового состояния на практике реализуется через дискретные унитарные операции — вентили или гейты. Квантовые гейты отличаются от классических тем, что преобразовывают сразу все возможные состояния кубита, тогда как окончательное состояние кубита определяется только в момент осуществления измерения.

Одними из основных вентилях являются гейт Адамара (в схемах H) и CNOT (в схемах CX). Гейт Адамара, представленный матрицей в формуле 6, создает состояние суперпозиции.

$$\hat{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (6)$$

Гейт CNOT (формула 7) — двухкубитный гейт для создания запутанности между кубитами. Он действует на пару кубитов: управляющий и целевой. Если управляющий $|0\rangle$, то целевой кубит остается неизменным; если управляющий кубит $|1\rangle$, то целевой инвертируется.

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (7)$$

Схема преобразования данных — карта признаков

При методе опорных векторов нужно определить преобразование пространства признаков. В квантовом случае такой функцией служит карта признаков (feature map).

Каждый классический вектор признаков $x = (x_1, x_2, \dots, x_n)$, полученный на предыдущем этапе, преобразуется в чистое квантовое состояние $|\Phi(x)\rangle\langle\Phi(x)|$ с использованием нелинейного отображения, называемого квантовой картой признаков, что послужит фиксированным пространством признаков. Эти состояния представляют собой векторы в гильбертовом пространстве.

Существует множество методов перевода входных классических данных в квантовую форму. Один из наиболее известных — квантовое частотное кодирование, при котором каждый вектор признаков ассоциируется с определённым квантовым состоянием в зависимости от его частотных характеристик.

В рамках данной работы используются схемы отображения признаков первого порядка (ZFeatureMap) и второго порядка (ZZFeatureMap), основанные на расширении по операторам Паули [9]. Эти схемы карт признаков трудно моделировать с помощью классических компьютеров, но в то же время, поскольку они имеют небольшую глубину, их можно успешно реализовать на имеющихся устройствах NISQ.

Рассмотрим квантовую карту признаков. Обучение и классификация с использованием обычных опорных векторов (SVM) эффективны, когда внутренние произведения между векторами признаков могут быть вычислены эффективно. Например, классификатор с разделимыми состояниями (состояния, которые могут быть представлены как произведение отдельных состояний для каждого кубита) часто может быть эффективно реализован классически. Например, если у нас есть два кубита, общее состояние — тензорное произведение (product state) состояний кубитов — может выглядеть

как $|0\rangle \otimes |1\rangle$, что означает, что первый кубит находится в состоянии $|0\rangle$, а второй — в состоянии $|1\rangle$. Состояния, представляемые тензорным произведением состояний отдельных кубитов, являются разделимыми или сепарабельными, такие кубиты не имеют квантовой запутанности. Чтобы получить преимущество перед классическими подходами, используя уникальные свойства квантовых систем, такие как запутанность или суперпозиция, нам нужно реализовать карту, основанную на схемах, которые трудно смоделировать классически. Поскольку квантовые компьютеры не ожидаются как классически симулируемые, существует длинный список (универсальных) семейств схем, из которых можно выбирать. Здесь мы используем схему, которая хорошо работает в наших экспериментах и не слишком глубока. Мы определяем карту признаков на n -кубитах и глубиной d , генерируемую унитарной операцией, представленной формулой 8 [9].

$$U_{\Phi(x)} = \prod_{i=1}^n \exp(ix_i Z_i) \quad (8)$$

где: Z_i — оператор Паули на i -м кубите.

В ZFeatureMap применяются гейты Адамара (H) на каждый кубит для создания суперпозиции. Затем следуют вращения $R_z(x_i) = \exp(ix_i Z_i)$, кодирующие значение признака x_i в фазу кубита. В результате ZFeatureMap кодирует каждый признак x_i независимо, создавая линейное отображение данных.

Формула ZZFeatureMap учитывает как одиночные, так и парные взаимодействия (формула 9).

$$U_{\Phi(x)} = \prod_{i=1}^n \exp(ix_i Z_i) \prod_{i < j} \exp(i(\pi - x_i)(\pi - x_j) Z_i Z_j) \quad (9)$$

где: $Z_i Z_j$ — парный оператор Паули, моделирующий взаимодействие между i -м и j -м кубитами.

Схема ZZFeatureMap начинается с гейтов Адамара (H) для создания суперпозиции. Затем применяется вращение $R_z(x_i)$ для каждого кубита (как в ZFeatureMap). После добавляются двухкубитные гейты R_{zz} , кодирующие нелинейные взаимодействия $(\pi - x_i)(\pi - x_j)$. В результате создается нелинейное отображение, учитывающее корреляции между признаками.

Реализация QSVC

QSVC использует квантовое ядро (формула 10) вместе с классическим SVM [10].

$$K(x_i, x_j) = |\langle \Phi(x_i) | \Phi(x_j) \rangle|^2 \quad (10)$$

где: $\Phi(x)$ — квантовое состояние, создаваемое схемой кодирования данных.

Данные кодируются с помощью унитарных операций (карт признаков), затем считается точность воспроизведения одного состояния другим (fidelity). Для чистых состояний точность воспроизведения одного состояния другим считается по формуле 9, что схоже с формулой 11.

$$F(|\psi\rangle\langle\psi|, |\phi\rangle\langle\phi|) = |\langle\psi|\phi\rangle|^2 \quad (11)$$

QSVM функционирует аналогично классическому SVM, используя двойственную формулировку (формула 5, но ядро из формулы 10), но с заменой классического ядра на квантовое. Задача сводится к нахождению разделяющей гиперплоскости для квантовых векторов состояний $|\phi(x_i)\rangle$ в соответствии с их метками.

PegasosQSVC

Альтернативной реализацией QSVC, представленной в Qiskit, является PegasosQSVC. В отличие от двойственной задачи оптимизации в классическом QSVC, PegasosQSVC использует стохастический градиентный спуск (формула 12), что снижает вычислительную сложность с $O(n^2)$ для QSVC до $O(n)$, n — количество признаков. Благодаря этому алгоритм эффективнее для больших обучающих выборок [11].

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \max(0, 1 - y_i (w^T \phi(x_i) + b)) \quad (12)$$

Результаты

Выбран датасет CMC спама (Kaggle. Spam SMS Classification using NLP), коллекции публичного набора CMC-сообщений на английском языке, содержащих текст присланного сообщения и метки (спам или обычное сообщение), которую ставили пользователи, получившие данное сообщение. Он состоит из двух столбцов: «Class» — метка, определяющая категорию спам/не спам (spam/ham); «Message» — содержит текстовое сообщение на английском языке.

Предобработка, подготовка данных и обучение моделей

В рамках реализации фреймворка классический компьютер использовался для предварительной обработки данных, извлечения и уменьшения размерности признаков, а также для финальной интерпретации результатов. Квантовая часть — симулятор квантового компьютера — применялась для кодирования обучающей выборки, построения пользовательской карты признаков, обучения квантовой модели, оптимизации параметров и последующей обработки квантового вывода.

На первом этапе работы данные проверяются на пропущенные значения и дубликаты. Затем метки кодируются

с помощью LabelEncoding, метка «0» — не спам, метка «1» — спам. Данных со спамом 1032, без спама 4127.

Затем обработанные текстовые данные очищаются от лишних символов, приводятся к нижнему регистру и токенизируются. После этого из токенов удаляются цифры, знаки пунктуации, а также стоп-слова, что позволяет сосредоточиться на значимых словах.

Далее очищенные текстовые данные конвертируются в векторное представление с помощью SBERT. В данной работе используется модель трансформеров предложений paraphrase-multilingual-MiniLM-L12-v2.

Затем размерность уменьшается с помощью UMAP до 12 компонентов.

После векторизации данные масштабируются в диапазоне $[0, \pi]$ с помощью MinMaxScaler, что делает их более подходящими для подачи на вход квантовым алгоритмам.

Итогом этого этапа становится векторизованный набор данных, который готов для дальнейшей обработки. Затем он делится на тренировочные и тестовые данные с использованием функции train_test_split() из библиотеки sklearn. В тренировочной части данных распределение по классам: класс 0—3607 объектов, класс 1—520 объектов; в тестовой: класс 0 — 911 объектов, класс 1 — 121 объект.

Также стоит рассмотреть полный набор данных без уменьшения размерности. Предобработка полного датасета аналогична, но без использования UMAP.

Классическая классификация

Несмотря на то, что SVM в основе работает с метками $\{-1; +1\}$, а имеется $\{0; 1\}$, внутри своих алгоритмов имеющиеся классы отображаются в $\{-1; +1\}$.

На данных, полученных на предыдущем этапе (набор с размерностью 384 признака и данных, уменьшенных до 12 признаков), обучена модель классического машинного обучения SVC, которая затем тестировалась на тестовой выборке и оценивалась по метрикам: accuracy, precision, recall, f1-мера.

Квантовая классификация

Предобработанные данные сопоставляются с квантовыми состояниями в гильбертовом пространстве с помощью карты признаков.

Обучение происходит на AerSimulator.

Для реализации QSVC ядро для классификатора опорных векторов строится из квантовых состояний, созданных квантовой картой признаков.

Ядро используется для обучения и тестирования модели QSVС. То же оптимизированное ядро используется при классификации тестовых данных.

Для модели QSVС используется меньший набор данных, состоящий из 400 случайно выбранных объектов из предобработанного датасета для обучения и 200 случайно выбранных объектов для тестирования. Он рассматривается отдельно, так как он требует большого количества оперативной памяти, из-за этого сравнить его с другими алгоритмами при прочих равных условиях не имеется возможности.

PegasosQSVС обучается и тестируется на полном наборе данных с 12 признаками.

Завершающая обработка

После выполнения измерений полученные квантовые выходные данные преобразуются в числовые значения, отражающие вероятностное распределение по классам. Эти значения интерпретируются как предсказания модели относительно принадлежности каждого текстового фрагмента к тому или иному классу. Предсказанные метки сравниваются с фактическими метками из тестовой выборки, далее оценивается качество модели по метрикам: точность (accuracy), полнота (recall), точность (precision) и f1-мера.

Обсуждение полученных результатов

Результаты классического SVC с набором данных с большой размерностью и SVC с PegasosQSVС с одинаковым набором данных размерности 12 представлены в таблице 1. Теоретическая сложность SVC $O(n^3)$, QSVС $O(n^2)$, PegasosQSVС $O(n)$, n — количество точек данных.

Как видим, результаты квантового PegasosQSVС превзошли ненамного классический SVC при равных условиях (на 12 признаках).

Результаты QSVС представлены в таблице 2.

Полученные результаты для QSVС сравнимы с SVC для одинаковой выборки, однако потенциал данного ал-

Таблица 1

Результаты SVC, PegasosQSVС

Метрика Алгоритм	Время выполнения fit, с	accuracy	precision	recall	f1-мера
SVC (scikit-learn) с сохраненной размерностью данных	20,06	0,975	0,944	0,835	0,886
SVC (scikit-learn) 12D	0,61	0,937	0,769	0,661	0,711
PegasosQSVС 12D	0,51	0,96	0,864	0,785	0,823

Таблица 2

Результат работы SVC, PegasosQSVС

Метрика Алгоритм	Время выполнения fit, с	accuracy	precision	recall	f1-мера
SVC	0,04	0,965	0,792	0,905	0,844
QSVС	9105	0,955	0,875	0,667	0,757

горитма рассмотреть на полном наборе данных не имеется возможности.

Следует отметить ограничения, не учтенные в данной работе: эксперименты на реальных квантовых компьютерах затруднены из-за аппаратных ограничений современных устройств, включая ограниченный объем памяти NISQ-устройства; в момент написания работы отсутствия поддержки необходимых вентилей (гейт Адамара, гейт CNOT), а также миграции IBM Quantum (главного поставщика реального квантового устройства) на другую платформу 01 июля 2025 года.

Несмотря на вышеуказанные ограничения, данная работа может послужить основой для дальнейших исследований внедрения квантовых алгоритмов в обработку естественного языка и другие задачи машинного обучения.

ЛИТЕРАТУРА

1. Kanev A.I., Savchenko G.A., Grishin I.A., Vasiliev D.A., Duma E.M. Sentiment analysis of multilingual texts using machine learning methods // 2022 International Conference on Engineering and Telecommunication (EnT). — 2022.
2. Tsang I.W., Kwok J.T., Cheung P.M. Core Vector Machines: Fast SVM Training on Very Large Data Sets // Journal of Machine Learning Research. — 2005. — Т. 6.
3. Santi P., Mishra K., Mohanty S. Quantum Text Classifier — A Synchronistic Approach Towards Classical and Quantum Machine Learning [Электронный ресурс]. — 2023. — URL: <https://doi.org/10.48550/arXiv.2305.12783>
4. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [Электронный ресурс]. — 2018. — URL: <https://doi.org/10.48550/arXiv.1810.04805>
5. A.I. Kanev and V.I. Terekhov, Evaluation issues of query result ranking for semantic search / Information technologies, telecommunications, and control systems (ITICS2020), 2020 SCOPUS

6. Reimers N., Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks [Электронный ресурс]. — 2019. — URL: <https://doi.org/10.48550/arXiv.1908.10084>
7. McInnes L., Healy J., Melville J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction [Электронный ресурс]. — 2018. — URL: <https://doi.org/10.48550/arXiv.1802.03426>
8. Nielsen M.A., Chuang I.L. Quantum Computation and Quantum Information. — Cambridge: Cambridge University Press, 2010. — 39 с.
9. Havlicek V., Corcoles A.D., Temme K., Harrow A.W., Kandala A., Chow J.M., Gambetta J.M. Supervised learning with quantum enhanced feature spaces [Электронный ресурс]. — 2018. — URL: <https://doi.org/10.48550/arXiv.1804.11326>
10. Schuld M. Supervised quantum machine learning models are kernel methods [Электронный ресурс]. — 2021. — URL: <https://doi.org/10.48550/arXiv.2101.11020>
11. Sun J. Primal Estimated Subgradient Solver for SVM for Imbalanced Classification [Электронный ресурс]. — 2022. — URL: <https://doi.org/10.48550/arXiv.2206.09311>

© Бабасанова Надежда Сергеевна (bns22k060@student.bmstu.ru); Канев Антон Игоревич (aikanev@bmstu.ru);

Михеева Валерия Алексеевна (mikhееvava@bmstu.ru)

Журнал «Современная наука: актуальные проблемы теории и практики»