СОДЕРЖАНИЕ

Номер 6, 2022

АНАЛИЗ ДАННЫХ

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

ПАРАЛЛЕЛЬНОЕ И РАСПРЕДЕЛЕННОЕ ПРОГРАММИРОВАНИЕ

ТЕОРИЯ ПРОГРАММИРОВАНИЯ: ФОРМАЛЬНЫЕ МОДЕЛИ И СЕМАНТИКА

CONTENTS

No. 6, 2022

EDN: KWOQRH
ПРОГРАММИРОВАНИЕ, 2022, № 6, с. 3–13

АНАЛИЗ ДАННЫХ

УДК 004.421.6

ИСПОЛЬЗОВАНИЕ МЕТОДОВ ГЛУБОКОГО ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ ДЛЯ ОТБОРА ПРИЗНАКОВ СЕТЕВОГО ТРАФИКА ПРИ ОБНАРУЖЕНИИ КОМПЬЮТЕРНЫХ АТАК

© 2022 г. В. В. Беликов[a,*] (ORCID: 0000-0003-1423-1072)

[a] МИРЭА – Российский технологический университет
119333 Москва, проспект Вернадского, д. 78, Россия
**E-mail: belikov_v@mirea.ru*

Поступила в редакцию 28.04.2022 г.
После доработки 21.06.2022 г.
Принята к публикации 07.07.2022 г.

В статье предложено решение задачи отбора признаков сетевого трафика с использованием методов глубокого обучения с подкреплением, представляющее классификацию в виде последовательного процесса, на каждом шаге которого принимается решение о достаточности наличия имеющихся значений признаков для соотнесения объекта с классом. Предложенное решение позволяет варьировать количество используемых признаков от одного экземпляра к другому. Проведенный эксперимент продемонстрировал возможность использования такого решения для увеличения обобщающей способности моделей классификации и снижения переобучения при их использовании в СОВ сетевого типа для обнаружения компьютерных атак, в том числе при наличии только несбалансированных обучающих наборов данных.

DOI: 10.31857/S0132347422060024

1. ВВЕДЕНИЕ

Компьютерные системы оказывают все большее влияние на современную жизнь, что делает кибербезопасность важной областью исследований. Среди различных инструментов обеспечения защиты компьютерных сетей как одного из основных компонентов компьютерных систем ключевую роль играют системы обнаружения вторжений (intrusion detection systems, СОВ). Недостаточная эффективность применения сигнатурного анализа, в том числе его ограниченные возможности при обнаружении неизвестных ранее компьютерных атак, а также бурное развитие методов интеллектуального анализа данных являются причинами большого числа исследований, посвященных использованию альтернативного подхода, закладываемого в основе СОВ: подхода, основанного на методах машинного обучения. Вместе с тем, на успешное применение разработанных с использованием методов машинного обучения решений накладывает ограничения особенности предметной области обнаружения компьютерных атак в сети: отсутствие или ограниченный объем имеющихся наборов реальных данных; несбалансированная обучающая выборка; высокая вариативность сетевого трафика и постоянное совершенствование способов проведения компьютерных атак.

2. АНАЛИЗ И ПРОБЛЕМА

Проблема отбора признаков сетевого трафика для обнаружения компьютерных атак. Задачу обнаружения компьютерных атак в контексте применения методов машинного обучения чаще всего представляют как задачу бинарной классификации объекта, извлеченного из имеющихся данных. СОВ сетевого типа используют в качестве источника данных сетевой трафик, в основе которого лежат пакеты. Пакеты являются основными единицами сетевого взаимодействия и представляют из себя оформленные блоки данных, состоящие из служебной информации (например, флаг TCP/UDP) и полезной нагрузки (payload), формат которой определяется используемым протоколом передачи данных. Чаще всего при этом объектом классификации выступает поток, который является набором пакетов, отражающим сетевую среду в течение определённого интервала времени. Однако бинарный формат и большой объем пакетов в рамках одного потока делает невозможным их применение в методах машинного обучения напрямую, что обуславливает необходимость предварительного конструирования вектора признаков, которые отражают либо статистические свойства потока (например, доля флагов TCP, средняя длина полезной нагрузки), либо свойства потока как последовательности пакетов;

наиболее распространенным является первый подход, исследованию которого и посвящена данная статья. Существует несколько практических инструментов, позволяющих решать указанную задачу, например, Argus[1], CICFlowMeter[2], NFStream[3], Fullstats[4]. Общее количество признаков, извлекаемых указанными инструментами, может достигать 85. Однако использование всего набора признаков на этапе эксплуатации модели приводит к задержке реакции СОВ.

Кроме этого, достаточно хорошо освещенной проблемой является недостаточное качество имеющихся обучающих выборок, выражающееся в наличии следующих недостатков:

• ввиду отсутствия реальных наборов данных, не выкладываемых из соображений приватности, использование обучающих выборок приводит к переобучению модели, к ее слабой адаптации для применения в отношении сетевого трафика, по своим характеристикам отличающегося от используемого для обучения модели;

• несбалансированность выборки, подавляющая часть которой является обычным трафиком, усложняет обучение, характеризующееся большим количеством ошибок второго рода.

Высокая вариативность сетевого трафика и постоянное совершенствование способов проведения компьютерных атак исключают возможность по созданию универсальной обучающей выборки, устраняющей вышеперечисленные недостатки, и обосновывают необходимость увеличения обобщающей способности моделей классификации при их использовании в СОВ сетевого типа. Одним из подходов для решения указанной проблемы является отбор признаков сетевого трафика.

Математическая постановка задачи. Задача классификации сетевого трафика для обнаружения компьютерных атак с использованием СОВ может быть формализована следующим образом. Пусть задано множество объектов сетевого трафика $\mathcal{X}$, представленных вектором из d признаков $\mathbf{x} = (x_1,\ldots,x_d)$, множество классов $\mathcal{Y} = \{0$ – безопасный сетевой график, 1 – компьютерная атака$\}$ и множество моделей (гипотез) $\mathcal{H}$, описываемых в виде функции, которая каждому объекту ставит в соответствие один из классов $h : \mathcal{X} \to \mathcal{Y}$. Тогда истинная ошибка модели $h \in \mathcal{H}$ определяется как неотрицательная функция потерь $\ell : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$. Для бинарной классификации функция потерь задается равной единице, если гипотеза неправильно определила класс, и нулю в ином случае:

$$\ell(h(\mathbf{x}), y) = \begin{cases} 1, & \text{если } h(\mathbf{x}) \neq y \\ 0, & \text{иначе} \end{cases} \tag{2.1}$$

При заданном вероятностном распределении $\mathcal{D}$ над $\mathcal{X} \times \mathcal{Y}$ функция ℓ является случайной, а ее математическое ожидание именуется истинной ошибкой (true risk) модели h [1]:

$$L_D(h) = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}}[\ell(h(\mathbf{x}), y)]$$

Так как в большинстве случаев распределение $\mathcal{D}$ неизвестно, то модель выбирается по имеющейся выборке из m объектов $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}^m, y^m))$ на основе минимизации эмпирической ошибки (empirical risk), рассчитываемой как среднее арифметическое наблюдаемых значений функции потерь ℓ:

$$L_S(h) = \frac{1}{m}\sum_{i=1}^{m}\ell(h(\mathbf{x}^i), y^i) \tag{2.2}$$

Вместе с тем, использование вместо истинной ошибки эмпирической в качестве целевого показателя при решении задачи классификации ввиду ограниченного размера обучающей выборки часто приводит к переобучению. Для численной оценки переобучения может быть использовано взятое в отношении случайной выборки S математическое ожидание ошибки обобщения, определяемое как разница между истинной и эмпирической ошибкой модели h [2]:

$$L_{\mathcal{D}}(h) - L_S(h) \tag{2.3}$$

Различным семействам моделей H соответствуют различные значения $\sup_{h\in H}\left|L_{\mathcal{D}}(h) - L_S(h)\right|$, определяемыми характеристикой этого семейства, например, VC-размерность [3]. Одним из наиболее известных и теоретически изученных подходов, используемых для уменьшения энтропии, является минимизация структурного риска (structural risk minimization), при котором для набора вложенных семейств моделей $H_1 \subset H_2 \subset \ldots \subset H_r \subset$ ищется компромисс между сложностью модели и эмпирической ошибкой за счет добавления регуляризационной функции $F(h)$, описывающей сложность семейства, к которому эта модель принадлежит:

$$\arg\min_h L_S(h) + F(h) \tag{2.4}$$

Набор семейств $\{H_r\}$, $H_1 \subset H_2 \subset \ldots$, при имеющейся возрастающей последовательности положительных чисел $a_1 < a_2 < \ldots$ может быть получен заданием H_r как множества моделей, среднее количество используемых признаков которых меньше или равно a_r, а в качестве характеристики сложности H_r, используемой в регуляризационной функции $F(h)$ – значение a_r.

[1] https://openargus.org/
[2] https://github.com/ahlashkari/CICFlowMeter
[3] https://www.nfstream.org
[4] https://www.cl.cam.ac.uk/research/srg/netos/projects/brasil/

Для использования в таком подходе модель должна быть расширена до набора двух функций:

$$h = (h_y, h_z), \quad h_y : \mathcal{X} \to \mathcal{Y}, \quad h_z : \mathcal{X} \to \mathcal{Z} \qquad (2.5)$$

Здесь функция h_y ставит в соответствие объекту $\mathbf{x}$ предполагаемый класс y, а функция h_z – вектор $\mathbf{z} \in \mathcal{Z} = \{0,1\}^m$, где i-й элемент вектора $z_i = 1$, если i-й признак использовался для предсказания класса y. Тогда выражение (2.4) принимает вид:

$$\begin{aligned} &\arg\min_h L_S(h_y) + \lambda \frac{1}{m}\sum_{i=1}^{m} \|h_z(\mathbf{x}^i)\|_0 = \\ &= \arg\min_h \frac{1}{m}\sum_{i=1}^{m} [\ell(h_y(\mathbf{x}^i), y^i) + \lambda \|h_z(\mathbf{x}^i)\|_0] \end{aligned} \qquad (2.6)$$

Варьирование значения λ позволяет выбирать компромисс между ограничением среднего количества используемых признаков и эмпирической ошибкой классификации. Таким образом, это позволяет с использованием отбора наиболее значимых признаков повышать обобщающую способность модели без увеличения размера обучающей выборки m в системах обнаружения вторжений.

Еще одной проблемой описываемой предметной области является несбалансированность классов, которая заключается в том, что в имеющейся выборке, как правило, количество объектов сетевого трафика, соответствующих компьютерным атакам, значительно меньше количества объектов безопасного сетевого трафика $|\{(\mathbf{x}^i, y^i) \in S : y^i = 1\}| \ll$ $\ll |\{(\mathbf{x}^i, y^i) \in S : y^i = 0\}|$. Модели, построенные на основе такой выборки с использованием алгоритма, предназначенного для обучения на сбалансированном наборе, характеризуются критически маленьким значением полноты, так как с большей вероятностью относят новые наблюдения к классам, представленным большим числом обучающих примеров. Одним из наиболее известных подходов, позволяющим обойти описанную проблему без изменения пропорций классов в имеющейся выборке является классификация с использованием издержек [4]. При таком подходе применяется иная функция потерь, которая каждому виду ошибки классификации: первого и второго рода – ставит в соответствие стоимости: $C_{10} \in \mathbb{R}_+$ и $C_{01} \in \mathbb{R}_+$.

$$\ell(h_y(\mathbf{x}), y) = \begin{cases} C_{01}, & \text{если } h_y(\mathbf{x}) = 0 \text{ и } y = 1 \\ C_{10}, & \text{если } h_y(\mathbf{x}) = 1 \text{ и } y = 0 \\ 0, & \text{иначе} \end{cases} \qquad (2.7)$$

Функция потерь вида (2.7) является расширением функции потерь вида (2.1) и может быть сведена к ней при выполнении условия $C_{01} = C_{10} = 1$. При известных значениях C_{01} и C_{10}, определяемых предметной областью, cost-sensitive learning может использоваться для решения задач с несбалансированным набором данных. Если эти значения неизвестны, то они могут быть заданы пропорционально объему экземпляров каждого из классов в имеющейся выборке S:

$$\begin{aligned} C_{01} &= \frac{|\{(\mathbf{x}^i, y^i) \in S : y^i = 0\}|}{m} \\ C_{10} &= \frac{|\{(\mathbf{x}^i, y^i) \in S : y^i = 1\}|}{m} \end{aligned} \qquad (2.8)$$

Таким образом, поставленной в статье задачей является нахождение модели, которая является решением оптимизационной задачи (2.6) в отношении функции потерь вида (2.7)

3. ОБЗОР РАБОТ ПО ТЕМАТИКЕ ПРОЕКТА

В работе [5] представлен обзор существующих методов отбора признаков сетевого трафика для систем обнаружения вторжений; осуществлено их обобщение в три категории: фильтры (filter), основанные на показателях, не зависящих от метода классификации; методы обертки (wrapper), где значимость признаков основывается на результатах применения методов классификации для их разных комбинаций; гибридные методы. Проведено сравнение эффективности и производительности этих методов с использованием набора данных KDD 1999, для оценки использовались показатели полнота (Recall)

$$\text{Recall} = \frac{TP}{TP + FN}$$

и доля ложноположительных результатов (FPR)

$$FPR = \frac{FP}{FP + TN}$$

Здесь TP – количество истинно положительных результатов, TN – количество истинно отрицательных результатов, FP – количество ложноположительных результатов, FN – количество ложноотрицательных результатов. Сделан вывод об обоснованности дальнейшего развития методов отбора признаков сетевого трафика для систем обнаружения вторжений.

В работе [6] рассмотрено применение искусственных нейронных сетей в системах обнаружения вторжений; в основу отбора признаков сетевого трафика было положено использование искусственной нейронный сети с добавленным зашумленным узлом, где для оценки значимости признака использовалось значение показателя отношения сигнал-шум (signal-to-noise ratio, SNR):

$$SNR_i = 10 \log_{10}\left(\frac{\sum_{j=1}^{J} (w_{itp,j}^1)^2}{\sum_{j=1}^{J} (w_{N,j}^1)^2}\right)$$

Таблица 1. Параметры первого правила Snort для обнаружения атаки DNS spoofing

Имя параметра правила	Значение параметра правила
flow	to_client
content	|85 80 00 01 00 01 00 00 00 00|
content	|C0 0C 00 0C 00 01 00 00 00|<|00 0F|
fast_pattern	only

Здесь SNR_i – значение отношения сигнал–шум для i-го признака, J – количество узлов в скрытых слоях, $w^1_{i,j}$ – значения веса связи первого слоя между узлом i и узлом j, $w^1_{N,j}$ – значения веса связи первого слоя между зашумленным узлом N и узлом j. Экспериментальная оценка проведена с использованием сбалансированной выборки, сформированной на основе набора данных CDX.

В работе [7] статистические признаки сетевого трафика были обогащены биграммами, извлеченными из полезной нагрузки. Для отбора признаков сетевого трафика разработан метод рекурсивного добавления признаков (RFA) при обучении SVM – такой метод отличается от методов обертки тем, что отбор признаков происходит непосредственно на этапе обучения классификатора. Проведен эксперимент с использованием набора данных ISCX 2012.

В работе [8] предложен метод отбора признаков CFS-BA, представляющий из себя комбинацию основанного на корреляции фильтра (CFS) и метаэвристического алгоритма глобальной оптимизации Bat. Проведен эксперимент на наборах данных NSL-KDD, AWID, and CICIDS2017, который позволил авторам сделать вывод о повышении значений метрик-curacy и F-меры без повышения значения FPR при применении предложенного метода отбора признаков, где:

$$F_1 = 2\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Precision} = \frac{TP}{TP + FN}$$

Наконец в работе [9] для формирования признакового пространства оценка значимости и отбор признаков набора данных CICIDS2017 использовался энтропийный подход с последующим применением корреляционного анализа, что позволило количество признаков сократить с 84 до 10. Для устранения дисбаланса классов применялся метод занижения доли объектов определенного класса случайным сэмплированием (undersampling). Было осуществлено сравнение различных методов классификации в части обнаружения web-атак, по результатам которого был выбран RandomForest. Полученные результаты апробации обученной с использованием RandomForest модели на сетевом траффике, собранным авторами в реальной сети, продемонстрировали крайне низкие значения F-меры, несмотря на высокие значения показателей классификации для тестовой выборки из набора CICIDS2017. На основании полученных результатов авторами был сделан вывод о влиянии отличий в физической структуре сетей и настройках оборудования на возникновение ошибок классификатора и точность модели.

Однако в исследованиях, проводимых на данную тему, целью являлся отбор признаков для всего набора данных, что может быть рассмотрено как задача нахождения условного экстремума функции (2.2) с учетом ограничения на количество используемых признаков. Вместе с тем, ввиду как большого различия между разными видами сетевых атак, так и значимого разнообразия реализаций одного конкретного вида, для обнаружения вторжения с достаточным уровнем значения показателя эффективности классификации для каждого отдельного экземпляра сетевого трафика количество используемых признаков может значительно варьироваться. Так, для системы обнаружения вторжений Snort[5] в отношении одного и того же вида атаки – DNS spoofing – существует два правила обнаружения, каждое из которых является достаточным для сигнализирования о соответствующей атаке. При этом для выполнения первого достаточно найти в любом месте полезной нагрузки одного из пакетов входящего сетевого потока два шаблона выражения (табл. 1), тогда как для второго правила требуется нахождения трех шаблонов выражений, на расположение которых внутри полезной нагрузки накладываются ограничения вроде смещения относительно начала нагрузки (offset), допустимого расстояния между шаблонами (distance) и соответствия значения отдельных байтовых полей заданным тестам(byte_test) (табл. 2).

Решение оптимизационной задачи, приведенной в выражении (2.6), сокращает именно среднее, а не максимальное количество признаков сетевого трафика, что позволяет при необходимости ограничиваться небольшим количеством признаков для простых объектов классификации и использовать большее количество признаков для сложных.

Также в проведенных исследованиях проблема дисбаланса классов решалась с использованием методов занижения или завышения (oversampling) доли объектов определенного класса. В отличие от этого, использование функции ошибки, приведенной в выражении (2.7), позволяет решать напрямую проблему дисбаланса классов без изменения состава обучающей выборки.

[5] https://www.snort.org/

В работе [10] было впервые предложено использовать для отбора признаков целевую функцию вида (2.6). Для нахождения ее решения использовался алгоритм Q-learning с линейной аппроксимацией, что сделало применение предложенного метода ограниченным.

В работе [11] указанная целевая функция использовалась для решения задачи классификации с признаками, добывание которых требует затрат (classification with costly features); линейная аппроксимация в методе Q-learning была заменена полносвязной нейронной сетью. Однако в указанной работе не рассматривался вопрос о связи сокращения среднего количества используемых признаков с переобучением. Кроме этого, для эксперимента использовались сбалансированные наборы данных и для его оценки применялся показатель *Accuracy*, что делает проблематичными перенос полученных результатов на несбалансированные наборы данных.

Таблица 2. Параметры второго, альтернативного правила Snort для обнаружения атаки DNS spoofing

Имя параметра правила	Значение параметра правила
flow	to_client
content	|81 80|
depth	4
offset	2
byte_test	2,>,0,0,relative,big
byte_test	2,>,0,2,relative,big
content	|00 00 00 00|
within	–4
distance	4
content	|C0 0C 00 01 00 01|
distance	0
byte_test	4,<,61,0,relative,big
byte_test	4,>,0,0,relative,big

4. МЕТОДОЛОГИЯ ИССЛЕДОВАНИЯ

Нахождение решения, представленного в виде (2.5), требует представления процесса классификации объекта сетевого трафика в виде последовательного принятия решения, где на каждом шаге необходимо выбрать либо запрос дополнительного, ранее неизвестного модели значения признака, либо соотнесение с классом. Тогда такой процесс может быть описан как взаимодействие классификатора, выступающего в роли агента, с эпизодическим марковским процессом принятия решений, который задается набором следующих элементов:

• Множество состояний (наблюдений состояния) среды $s \in \mathcal{S}$, описываемых конкатенацией двух векторов:

$$s = (\mathbf{x} \| \mathbf{z}) \tag{4.1}$$

Здесь вектор $\mathbf{x} \in \mathcal{R}^d$ представляет из себя значения известных признаков; вектор $\mathbf{z} \in \{0,1\}^d$ определяет, какие признаки известны классификатору ($z_i = 1$, если i-й признак известен классификатору, $z_i = 0$ иначе).

• Множество доступных действий агента $a \in \mathcal{A}$, которое включает в себя множество действий на добывание значения признака (выбор признака, значение которого будет запрашиваться у среды) и множество действий классификации.

• Распределение вероятностей перехода на шаге t в состояние s' и получения награждения r при выполнении действия a в состоянии s, заданных на множестве $\mathcal{S} \times \mathcal{A}$:

$$p(s', r | s, a) = P[S^t = s', R^t = r | S^{t-1} = s, A^t = a]$$

При получении действия на добывание значения i-го признака среда возвращает состояние $s^{t+1} = (\mathbf{x} \| \mathbf{z})$, отличающееся от предыдущего состояния наличием значения x_i i-го признака, а также изменением значения z_i с 0 на 1. Возвращаемое вознаграждение $r_t = -\lambda$ при этом является фактически штрафом за использование дополнительного признака, величина которого определяется параметром λ из выражения (2.6).

При получении действия классификации y_{pred} среда переходит в терминальное состояние, обозначающее конец эпизода, а возвращаемое вознаграждение определяется в зависимости от истинной категории y_{true}:

$$r_t = -l(y_{\text{pred}}, y_{\text{true}}) \tag{4.2}$$

Действия агента выбираются в соответствии со стохастической стратегией, задаваемой условным распределением, определяющим вероятность действия a при условии нахождения в состоянии s:

$$\pi(a|s) = P(A^t = a | S^{t-1} = s)$$

Так как пространство состояний может быть большим или бесконечным, то вместо табличного метода представления используется представление в виде параметризованной функции $\pi : \Theta \times \times \mathcal{S} \times \mathcal{A} \to [0,1]$, задающей семейство функций $\{\pi_\theta\}_{\theta \in \Theta}$, таких что $\sum_{a \in \mathcal{A}} \pi_\theta(a|s) = 1$ для каждого s. Как правило, для параметризации стратегий агента используются искусственные нейронные сети.

Реализацией эпизода управляемого марковского процесса является траектория

$$\tau = (s_0, a_1, s_1, r_1, a_2, \ldots, s_{T-1}, a_T, s_T, r_T)$$
$$a_t \sim \pi(\cdot|s_{t-1}), (s_t, r_t) \sim p(\cdot|s_{t-1}, a_t) \quad (4.3)$$

Стратегия агента π вместе с вероятностями перехода марковского процесса принятия решения задают вероятностное распределение на множестве траекторий. Траектория численно характеризуется суммой полученных наград $R(\tau) = \sum_{k=0}^{T} r_k$, являющейся случайной величиной.

Для оценки стратегии агента используется оценочная функция состояния (value function) $V : \mathcal{S} \to \mathbb{R}$, рассчитываемая как математическое ожидание суммы наград по всем возможным траекториям, начинающимся с состояния s:

$$V^{\pi}(s) = \mathop{\mathbb{E}}_{\tau \sim p,\pi}[R(\tau)|S_0 = s] \quad (4.4)$$

Одним из наиболее известных результатов динамического программирования является доказательство существования и единственности при определенных условиях оптимальной оценочной функции $V^*(s) = \max_{\pi} V_{\pi}(s)$ для всех s. В работе [10] показано, что при задании марковского процесса принятия решения указанным выше образом решением уравнения (2.6) является оптимальная стратегия агента π^*, соответствующая оптимальной оценочной функции $V^* = V^{\pi^*}$. Доказательство остается верным и при замене функции потерь вида (2.1), используемой в указанной работе, на функцию потерь вида (2.7).

В случае большого размера пространства состояний $\mathcal{S}$, а также когда неизвестны вероятности перехода $p(s', r|s, a)$, для нахождения оптимальной стратегии агента π^* используются методы обучения с подкреплением, для которых достаточно наличия среды или ее имитационной модели, в ответ на текущее состояние s и выбранное агентом действия a сэмплирующей новое состояние s' и награду r.

В работе [11] для нахождения оптимальной стратегии использовался алгоритм Deep Q-learning. Однако в основе этого алгоритма заложена стохастическая аппроксимация как метод нахождения решения уравнения оптимальности Беллмана, позволяющая на каждой итерации распространять обновление только на один шаг назад. Это ограничение делает метод Deep Q-learning непрактичным в отношении марковского процесса принятия решений с “разреженной” наградой, когда наибольшее награждение выдается в конце эпизода, что справедливо для описываемой предметной области, где сумма вознаграждений за эпизод $R(\tau)$ в большей степени зависит от корректности классификации, являющейся финальным действием эпизода. Существующая модификация Retrace(lambda) [12], направленная на обход указанного ограничения, позволяет распространить обновление только до тех пор, пока действия используемой для исследования стратегии совпадают с оптимальными действиями обучаемой “жадной” стратегии.

В текущей работе использовался PPO с обрезанной суррогатной целевой функцией (proximal policy optimization with clipped surrogate objective), являющийся современным представителем группы методов, которые напрямую оптимизируют стратегию (policy-based) [13]. Данная группа методов является методами online policy, в которых отсутствует различение между обучаемой стратегией и стратегией, используемой для исследования, что позволяет выбирать требуемую глубину обновления для каждой итерации и, как следствие, обходить проблему “разреженной награды”. Выбор именно алгоритма PPO обосновывался стабильностью его работы; малым числом гиперпараметров; несложностью в реализации при одновременном обеспечении уровня эффективности, соответствующего другим современным методам обучения с подкреплением.

При использовании PPO на каждой итерации k обновления параметризованной стратегии π_{θ_k} после выполненных заданного количества шагов взаимодействия со средой используется несколько итераций метода градиентного спуска для нахождения решения оптимизационной задачи со следующей суррогатной целевой функцией, рассчитываемой в применении к полученному набору $\mathcal{D}_k = \{\tau\}$ траекторий вида (4.3):

$$L^{CLIP}(\theta_k, \theta) = \frac{1}{|\mathcal{D}_k T|} \sum_{\tau \in \mathcal{D}} \sum_{t=0}^{T} L^{CLIP}_{\tau,t}(\theta) \quad (4.5)$$

$$L^{CLIP}_{\tau,t}(\theta) = \min(r_t(\theta)\hat{A}_t^{\pi_{\theta_k}}, \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)$$
$$r_t(\theta) = \frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} \quad (4.6)$$

В формуле (4.6) функция $\mathrm{clip}(x, x_{\min}, x_{\max})$ ограничивает элемент x отрезком допустимых значений $[x_{\min}, x_{\max}]$, задавая таким образом доверительную область, а ϵ выступает в роли гиперпараметра. Для расчета статистической оценки значения Advantage функции $A_t^{\pi} = Q^{\pi}(s_t, a_t) - V^{\pi}(s_t)$ для t-го состояния имеющейся траектории используется формула обобщенной оценки (Generalized Advantage Estimation – GAE), позволяющая с использованием гиперпараметров γ и μ варьировать и находить баланс между значениями смещения и дисперсии (bias/variance tradeoff).

$$\hat{A}_t^{(\gamma,\mu)} = \sum_{;=0}^{\infty} (\gamma\mu)^l \delta^V_{t+1}, \quad (4.7)$$

где

$$\delta_t^V = -V^{\pi_{\theta_k}}(s_t) + r_t + \gamma V^{\pi_{\theta_k}}(s_{t+1})$$

В свою очередь для расчета значения функции ценности $V^{\pi_{\theta_k}}$ используется также аппроксимация на основе нейронной сети V^θ (critic – "критик"), часто первые слои которой разделяют архитектуру и веса нейронной сети, используемой для аппроксимации стратегии агента. Обновление весов нейронной сети на каждой итерации k достигается минимизацией следующей функции

$$L^V(\theta) = \frac{1}{|\mathcal{D}_k T|}\sum_{\tau\in\mathcal{D}}\sum_{t=0}^{T}\left(V^\theta(s_t) - \sum_{i=t}^{T} r_i\right)^2 \qquad (4.8)$$

Выбор необходимой степени исследования новых для агента действий (exploration/explotation tradeoff) возможен максимизацией функции энтропии агента, который рассчитывается как статистическая оценка средней по всем посещаемым состояниям отрицательной энтропии стратегии агента

$$L^{ENT}(\theta) = \frac{1}{|\mathcal{D}_k T|}\sum_{\tau\in\mathcal{D}}\sum_{t=0}^{T}\sum_{a\in\mathcal{A}}\pi^\theta(a|s)\log\pi^\theta(a|s) \qquad (4.9)$$

Так как градиентный спуск является наиболее ресурсоемкой операцией, то в одной из самых популярных версий алгоритма PPO, представленной ниже, оптимизация выражений (4.5), (4.8), (4.9) происходит одновременно за счет их суммирования в единую целевую функцию с использованием соответствующих коэффициентов k^V и k^{ENT}, также выступающих в роли гиперпараметров.

Алгоритм 1. Алгоритм PPO с ограничивающей суррогатной целевой функцией

Входные данные: θ_0 – начальные параметры стратегии

Цикл $i = 0,1,2,\ldots$ **выполнять**

Получить для стратегии $\pi_k = \pi_{\theta_k}$ набор траекторий $\mathcal{D}_k = \{\tau\}$

Рассчитать значения $\hat{A}_t^{\pi_k}$ с использованием формулы (4.7)

Выполнить обновление параметров стратегии θ_k с использованием K шагов метода стохастического градиентного спуска

$$L(\theta) = L^{CLIP}(\theta_k,\theta) - k^V L^V(\theta) - k^{ENT} L^{ENT}(\theta)$$

$$\theta_{k+1} \leftarrow \arg\max_\theta L$$

Конец цикла

5. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТАЛЬНОГО АНАЛИЗА

Проведение эксперимента включало в себя два этапа:

• обучение с использованием алгоритма 1 модели и ее тестирование на наборе данных CIC-IDS2017 [14];

• апробация обученной модели в отношении реального траффика, используемого в работе [9].

При проведении эксперимента для обеспечения возможности апробации обученной модели в отношении реального траффика использовался набор данных CICIDS2017, в части нелегитимного сетевого траффика включающий только атаки грубого перебора (Brute Force), межсайтовый скриптинг (XSS) и Sql-иньекции, представляющие из себя наиболее известные примеры реализации уязвимостей веб-приложений [15]. Предобработка данных проводилась в порядке, указанном в [9], за исключением того, что дополнительно осуществлялись: выбор пересечения множеств признаков из набора CICIDS2017 и набора траффика, полученного на реальной сетевой инфраструктуре; исключение признаков с нулевой дисперсией ввиду их полной неинформативности; Z-нормализация оставшихся данных. После случайного перемешивания предобработанный набор был разделен на три выборки: обучающую (4096 объектов), валидационную (990 объектов) и тестовую (2181 объект). Общее количество признаков: 38.

Для проведения эксперимента была создана программная реализация формализованной выше среды марковского процесса принятия решения, совместимая с OpenAI gym. Исходный код среды и программного обеспечения, используемого для проведения эксперимента, доступны в репозитории

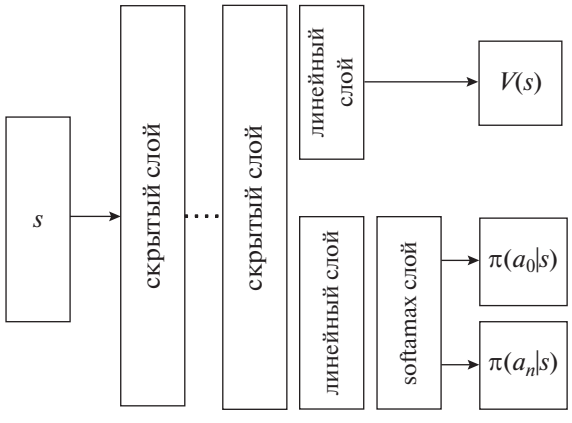

Рис. 1. Архитектура используемой нейронной сети.

Таблица 3. Найденные оптимальные значения гиперпараметров

Название гиперпараметра	Найденное оптимальное значение
Гиперпараметры градиентного спуска PPO	
Размер батча данных	1024
Коэффициент скорости обучения	0.0001 с линейным убыванием
Пороговое значение коэффициента нормирования градиента	1.0
Количество эпох	20
Гиперпараметры алгоритма PPO	
Количество шагов взаимодействия с каждой отдельной средой	128
Количество параллельно запущенных сред	16
Порог обрезания ε	0.6
Коэффициент k^V оптимизируемой функции "критика" L^V	0.8
Коэффициент k^{ENT} оптимизируемой функции энтропии L^{ENT}	0.075
Параметр длины горизонта μ	0.6
Гиперпараметры нейронной сети	
Количество скрытых слоев	3
Размер скрытого слоя	256

`https://github.com/james116blue`

Значения коэффициентов $C_{01} = 0.7$ и $C_{10} = 0.3$ выражения (2.7), определяющих в среде марковского процесса принятия решения награду за действие, соответствующее неправильной классификации, рассчитывались по формуле (2.8). Обучение производилось для следующих значений λ целевой функции (2.6), определяющих абсолютную величину отрицательного вознаграждения, выдаваемого средой в МППР при осуществлении агентом действия на добывания признака $r_i = -\lambda$: 0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001. Для каждого значения λ обучалось пять моделей, соответствующих разным случайным инициализациям (random seed), с использованием валидационной выборки отбиралась лучшая. Сбор траекторий для ускорения процесса обучения осуществлялся на 16 параллельных средах.

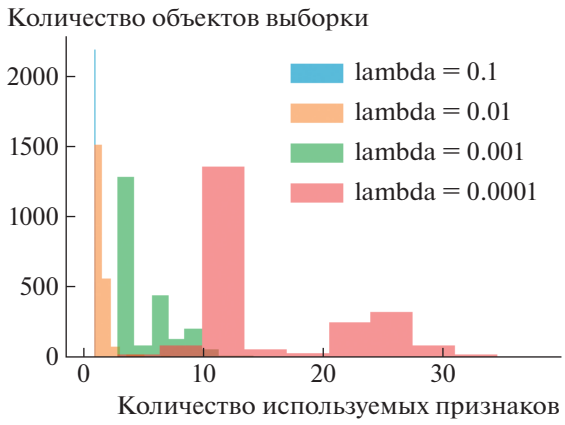

Рис. 2. Гистограмма количества используемых для классификации признаков.

Архитектура нейронной сети, используемой для аппроксимации стратегии и функции ценности, включала в себя несколько скрытых слоев, на первый из которых поступал вектор, описывающий состояние агента s (рис. 1). Так как на каждом шаге эпизода набор доступных действий зависел от того, значения каких признаков уже известны, то дополнительно перед слоем softmax использовалось маскирование недоступных действий $\mathbf{o} = \mathbf{u} - 10^6(0,0\|\mathbf{z})$, где $\mathbf{u}$ – выходные значения линейного слоя, находящегося перед softmax слоем, $\mathbf{o}$ – входные значения softmax слоя, $\mathbf{z}$ – вектор известных признаков состояния агента. Два нуля (0, 0), конкатенируемые с вектором $\mathbf{z}$, использовались для указания агенту о возможности выбора действий классификации. Коэффициент 10^6 давал для действий, соответствующим уже известным признакам ($z_i = 1$), нулевую вероятность $\pi(a_i|s) = e^{o_i}\left(\sum_j e^{o_j}\right)^{-1}$, обусловленную точностью чисел с плавающей точкой в программном пакете pytorch. Количество скрытых слоев и их размерность являлись гиперпараметрами модели. На каждом линейном слое в качестве функции активации использовалась ReLU $f(x) = \max(0, x)$.

Для выбора оптимальных значений гиперпараметров использовался применительно к обучающей выборке метод Tree-structured Parzen Estimator Approach (TPE) [16] с ранним окончанием обучения на основе медианных оценок значений

Таблица 4. Результаты эксперимента

Среднее количество используемых признаков	Точность	Полнота	F-мера
1	0.806	0.915	0.857
1	0.86	0.894	0.877
1.41	0.953	0.87	0.909
1.42	0.956	0.882	0.918
4.87	0.953	0.876	0.913
7.05	0.961	0.882	0.92
15.37	0.949	0.883	0.915

модели на валидационной выборке для коэффициента $\lambda = 0.01$ целевой функции (2.6). Полученные значения гиперпараметров представлены в таблице 3.

Полученные результаты оценки модели на тестируемой выборке набора данных CICIDS2017 представлены в табл. 4.

Как видно из полученных результатов, изменение значения λ позволяет варьировать между высоким значением эффективности классификации объектов и малым количеством используемых для этого признаков. При этом для любого значения λ имеет место следующее: для экземпляров, принадлежность к классу определить которых определяется более комплексной зависимостью, модель может запрашивать большее количество признаков. В отношении других экземпляров модели достаточно использования меньшего количества признаков. Это также продемонстрировано на гистограмме количества используемых признаков для каждого классифицируемого объекта – рис. 2. Например, для $\lambda = 0.0001$ максимальное количество используемых признаков может доходить до 40, тогда как модой является значение 10.

При этом для разных значений λ множество наиболее часто используемых признаков и их порядок могут отличаться, что демонстрируется на рис. 3, 4.

Так как в работе [9] сравнивались только модели малослойного (shallow) обучения, то на обучающей выборке набора данных CICIDS2017 также было осуществлено обучение глубокой полносвязной нейронной сети, в которой были реализованы следующие механизмы, направленные на минимизацию переобучения и повышения обобщающей способности модели:

- раннее прерывание обучения с использованием валидационной выборки [17];
- L2 регуляризация весов нейронной сети [18];
- случайное исключение отдельных нейронов (dropout) [19].

Параметры указанных механизмов, а также количество и размерность скрытых слоев и коэффициент скорости обучения являлись гиперпараметрами, значения которых подбирались на валидационной выборке. В итоге оценка модели на тестовой выборке CICIDS2017 указала значение показателя F-мера равным 0.94.

Результаты апробации модели, полученной с использованием алгоритма 1 на траффике реальной сетевой инфраструктуры, показали, что уменьшение среднего количества используемых

Рис. 3. Частота использования признаков для $\lambda = 0.0005$.

Рис. 4. Частота использования признаков для $\lambda = 0.0001$.

для классификации признаков приводит к снижению значения показателя F-меры, рассчитанной на выборке, полученной тем же порядком, что и обучающая (из набора данных CICIDS2017), но одновременно позволяет значительно повысить значение показателя F-меры, рассчитанной на выборке, по своим характеристикам отличающейся от обучающей. Об этом наглядно свидетельствует сравнение полученных значений F-меры с результатами применения метода Random Forest ($F_1 = 0.043$), описанных в статье [9], а также сравнение с результатом апробации глубокой полносвязной нейронной сети, показавшей себя немногим лучше модели, обученной с использованием алгоритма Random Forest ($F_1 = 0.075$) – рис. 5.

Таким образом, с использованием отбора только тех признаков, которые требуются для достижения компромисса между ограничением среднего количества используемых признаков и эмпирической ошибкой классификации, заданного в выражении 6, можно достичь повышения обобщающей способности модели и соответственно снижения эффекта переобучения. Это может быть объяснено в том числе тем, что повышение значения показателя F-меры достигается повышением точности за счет уменьшения полноты как одного из сопутствующих эффектов переобучения – рис. 6.

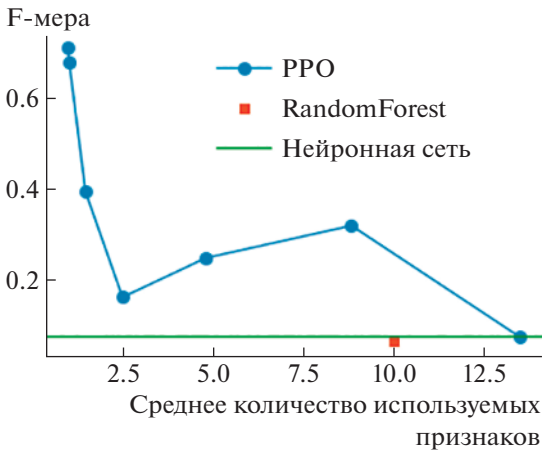

Рис. 5. Результаты апробирования моделей на выборке, полученной на реальной сетевой инфраструктуре.

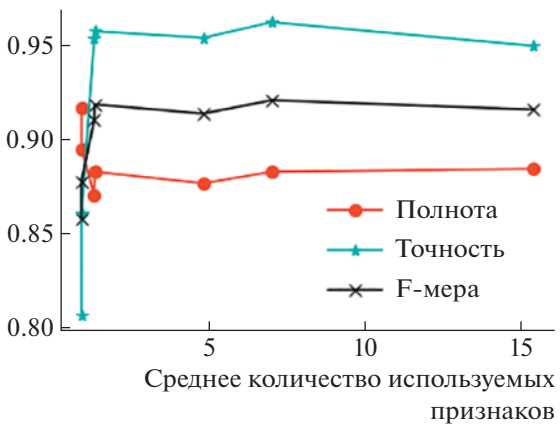

Рис. 6. Зависимость показателей классификации от среднего количества используемых признаков, оцененные по тестовой выборке набора данных CICIDS2017.

6. ВЫВОДЫ И ПРОДОЛЖЕНИЕ РАБОТЫ

Таким образом, в статье предложено решение задачи отбора признаков сетевого трафика с использованием методов глубокого обучения с подкреплением, представляющее классификацию в виде последовательного процесса, на каждом шаге которого принимается решение о достаточности наличия имеющихся значений признаков для соотнесения объекта с классом. Указанное решение позволяет варьировать количество используемых признаков от одного экземпляра к другому. Проведенный эксперимент продемонстрировал возможность использования такого решения для увеличения обобщающей способности моделей классификации и снижении переобучения при их использовании в СОВ сетевого типа для обнаружения компьютерных атак, в том числе при наличии только несбалансированных обучающих наборов данных.

СПИСОК ЛИТЕРАТУРЫ

1. *Shalev-Shwartz S., Ben-David S.* Understanding machine learning: From theory to algorithms. Cambridge university press. 2014. 445 p.
2. *Hardt M., Recht B., Singer Y.* Train faster, generalize better: Stability of stochastic gradient descent // International Conference on Machine Learning. 2016. P. 1225–1234.
3. *Vapnik V., Levin E., Cun Y.L.* Measuring the VC-Dimension of a Learning Machine // Neural Computation. 1994. V. 6. № 5. P. 851–8761.
4. *Ling C.X., Sheng V.S.* Cost-sensitive learning and the class imbalance problem // Encyclopedia of machine learning. 2011. P. 231–235.
5. *Lipmaa H., Yung M., Lin D.* Survey and Taxonomy of Feature Selection Algorithms in Intrusion Detection Syste // International Conference on Information Security and Cryptology. 2006. P. 153–167.
6. *Moore K.L., Bihl T.J., Bauer K.W.* Feature extraction and feature selection for classifying cyber traffic threats // The Journal of Defense Modeling and Simulation. 2017. V. 14. № 3. P. 217–231.
7. *Hamed T., Dara R., Kremer S.C.* Network intrusion detection system based on recursive feature addition and bigram technique // Computers & security. 2018. V. 73. P. 137–155.
8. *Zhou Y., Cheng G., Jiang S., Dai M.* Building an Efficient Intrusion Detection System Based on Feature Selection and Ensemble Classifier // Computer networks. 2020. V. 174. P. 107–123.
9. *Горюнов М.Н., Мацкевич А.Г., Рыболовлев Д.А.* Синтез модели машинного обучения для обнаружения компьютерных атак на основе набора данных CIC-IDS2017 // Труды Института системного программирования РАН. 2020. Т. 32. № 5. С. 81–94.
10. *Dulac-Arnold G., Denoyer L., Preux P., Gallinari P.* Datum-wise classification: a sequential approach to sparsity // InJoint European conference on machine learning and knowledge discovery in databases. 2011. P. 375–390.
11. *Janisch J., Pevny T., Lisy V.* Classification with costly features using deep reinforcement learning // InProceedings of the AAAI Conference on Artificial Intelligence. 2019. V. 33. P. 3959–3966.
12. *Hernandez-Garcia J.F., Sutton R.S.* Understanding multi-step deep reinforcement learning: a systematic study of the DQN target. arXiv preprint. arXiv:1901.07510. 2019.
13. *Schulman J., Wolski F., Dhariwal P., Radford A., Klimov O.* Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347. 2017.
14. Intrusion Detection Evaluation Dataset (CIC-IDS2017). https://www.unb.ca/cic/datasets/ids- 2017.htm. 2017.
15. *Лесько С.А.* Модели и сценарии реализации угроз для интернет-ресурсов // Russian Technological Journal. 2020. Т. 8. № 6. С. 9–33.
16. *Bergstra J., Bardenet R., Bengio Y., Kegl B.* Algorithms for hyper-parameter optimization. // Advances in neural information processing systems. 2011. V. 24. P. 123–145.
17. *Prechelt L.* Early stopping-but when? // InNeural Networks: Tricks of the trade. 1998. P. 55–69.
18. *Krogh A., Hertz J.* A simple weight decay can improve generalization // Advances in neural information processing systems. 1991. V. 4. P. 230–245.
19. *Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R.* Dropout: a simple way to prevent neural networks from overfitting // The journal of machine learning research. 2014. V. 15. № 1. P. 29–58.

EDN: DXYPRS
ПРОГРАММИРОВАНИЕ, 2022, № 6, с. 14–21

АНАЛИЗ ДАННЫХ

УДК 519.651

АЛГОРИТМ ВЫЧИСЛЕНИЯ КОРРЕКТНО ОКРУГЛЕННОГО ЗНАЧЕНИЯ ЭКСПОНЕНТЫ С ДВОЙНОЙ ТОЧНОСТЬЮ С ИСПОЛЬЗОВАНИЕМ АРИФМЕТИКИ РАСШИРЕННОЙ ДВОЙНОЙ ТОЧНОСТИ

© 2022 г. А. Н. Годунов[a,*] (ORCID: 0000-0001-5952-9185)

[a] *ФГУ Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук*
117218 Москва, Нахимовский просп., 36, корп. 1, Россия
*****E-mail: nkag@niisi.ras.ru*

Поступила в редакцию 01.06.2022 г.
После доработки 29.06.2022 г.
Принята к публикации 04.07.2022 г.

Использование функций, вычисляющих корректно округленное значение экспоненты позволяет, с одной стороны, получить наилучшее приближение, а с другой стороны, обеспечить переносимость (portability) программ. В статье предлагается алгоритм вычисления корректно округленного значения экспоненты, когда аргумент и значение функции являются числами двойной точности, а вычисления производятся с использованием чисел расширенной двойной точности. Дано формальное описание алгоритма. Представленный алгоритм реализован в виде функции на языке Си, проведено его тестирование и измерены временные характеристики. Проведено сравнение с другими алгоритмами.

DOI: 10.31857/S0132347422060036

1. ВВЕДЕНИЕ

Корректно округленное значение экспоненты есть результат округления математически точного значения экспоненты. Использование функций, вычисляющих корректно округленное значение экспоненты позволяет, с одной стороны, получить наилучшее приближение, а с другой стороны, обеспечить переносимость (portability) программ. Вычисление корректно округленного значения экспоненциальной функции рекомендовано стандартом IEEE 754 [1]. Если x – число двойной точности, не равное нулю, то e^x – трансцендентное число. В силу этого вычислить на компьютере точное значение экспоненты для ненулевого аргумента невозможно. Так как корректно округленное значение экспоненты – кусочно-постоянная функция, то можно ограничиться вычислением приближенного значения экспоненты. При этом погрешность вычисления экспоненты должна быть меньше, чем расстояние между точным значением экспоненты и ближайшей точкой разрыва функции округления.

Значения аргумента, для которых расстояние между точным значением экспоненты и ближайшей точкой разрыва функции округления минимально, считаются наиболее трудными (worst cases) для вычисления корректно округленного значения экспоненты. В. Лефевр, Ж.-М. Мюллер [2] были первыми, кто произвел поиск трудных случаев для вычисления корректно округленного значения экспоненциальной функции и других элементарных функций. В дальнейшем эта работа была продолжена В. Лефевром [3] и автором [4]. Результаты этих работ определили требования к точности вычисления экспоненты для получения корректно округленного значения этой функции.

В статье предлагается алгоритм вычисления корректно округленного значения экспоненциальной функции. Аргумент и значение функции – плавающие числа двойной точности, но при вычислениях используются плавающие числа расширенной двойной точности. Обычно для вычисления корректно округленного значения экспоненты с двойной точностью используется арифметика двойной точности. Единственным исключением, известным автору, является функция Лаутера (Lauter) [8], написанная на ассемблере и оптимизированная для процессора Intel Itanium. Эта функция использует числа расширенной двойной точности при вычислениях. Реализация предлагаемого алгоритма на языке Си (и поэтому мобильная) требует существенно меньше времени на вычисление корректно округленного значения экспоненты, чем функция Лаутера.

2. ПРЕДВАРИТЕЛЬНЫЕ СВЕДЕНИЯ

В этом разделе даны основные обозначения, определения и используемые результаты. Напомним, что для представления чисел двойной точности используется основание 2 и 53-битная мантисса. При вычислениях мы используем арифметику расширенной двойной точности: основание 2, длина мантиссы 64 бита.

Стандарт IEEE 754 [1] определяет следующие режимы округления: округление к ближайшему, округление к плюс бесконечности, округление к минус бесконечности и округление к нулю. Предусмотрено два вида режима округления к ближайшему: roundTiesToAway (к большему, если ближайших два) и roundTiesToEven (к числу с четной мантиссой, если ближайших два).

Предлагаемый алгоритм вычисляет корректно округленное значение e^x для всех режимов округления, но при вычислениях используется только округление к ближайшему (roundTiesToEven). Так как e^x всегда положительно, то округление к 0 эквивалентно округлению к минус бесконечности. В силу этого мы не будем рассматривать округление к 0. Если x – алгебраическое число не равное 0, то e^x – трансцендентное число. Поэтому при округлении e^x оба вида округления к ближайшему эквивалентны.

Определение 2.1. *Пусть X – арифметическое выражение, x – точное значение X, а x^* – вычисленное значение X. Величину $\Delta(x) = x^* - x$ будем называть абсолютной погрешностью вычисления.*

Погрешность вычисления обусловлена ошибками округления. Если x – действительное число, то через $\langle x\rangle$ будем обозначать округленное значение x в режиме округления к ближайшему (roundTiesToEven) при использовании арифметики плавающих чисел расширенной двойной точности.

Определение 2.2. *Пусть p и q – целые числа такие, что $p \ge q$. Множества $B^+_{p,q}$, $B^-_{p,q}$ и $B_{p,q}$ мы определяем следующим образом:*

- $B^+_{p,q}$ *– множество всех чисел вида*

$$x = b_p \cdot 2^p + b_{p-1} \cdot 2^{p-1} + \ldots + b_q \cdot 2^q,$$

где каждое из чисел $b_p, b_{p-1}, \ldots, b_q$ равно 0 *или* 1;

- $B^-_{p,q} = \{x | -x \in B^+_{p,q}\}$;

- $B_{p,q} = B^+_{p,q} \cup B^-_{p,q}$.

Определение 2.3. *Пусть n, m, p и q – целые числа такие, что $n \ge m > p > q$. Множество $T_{n,m,p,q}$ мы определяем следующим образом*:

$$T_{n,m,p,q} = \{(t_1, t_2, t_3) | t_1 \in B_{n,m}, t_2 \in B_{m-1,p},$$
$$|t_2| \le 2^{m-1}, t_3 \in B_{p-1,q}, |t_3| \le 2^{p-1}\}.$$

Определение 2.4. *Пусть n – целое, а x – действительное числа. Тогда $R_n(x)$ есть ближайшее к x число вида $m \cdot 2^n$, где m – целое. В случае неоднозначности выбираем четное m.*

Если n – целое, а x – действительное число, то

$$|R_n(x) - x| \le 2^{n-1}.$$

Если n – целое, а x – число расширенной двойной точности такие, что $|x| < 2^{n+62}$, и вычисление $x + 3 \cdot 2^{n+62}$ не приводит к переполнению, то

$$R_n(x) = \langle x + 3 \cdot 2^{n+62}\rangle - 3 \cdot 2^{n+62}.$$

Теорема 2.1 (алгоритм Fast2Sum [7]). *Пусть x и y – машинные числа такие, что $|x| \ge |y|$. Если их сложение не ведет к переполнению, то существуют такие машинные числа z и zz, что $z = \langle x + y\rangle$ и $x + y = z + zz$. Эти числа могут быть получены следующим образом:*

$$z := \langle x + y\rangle;$$
$$w := \langle z - x\rangle;$$
$$zz := \langle y - w\rangle.$$

Алгоритм Fast2Sum был предложен Деккером. Ниже мы будем использовать следующую нотацию для алгоритма Fast2Sum:

$$(z, zz) \leftarrow Fast2Sum(x, y).$$

Следующее определение введено В. Лефевром и Ж.-М. Мюллером (см. [2] и [7]).

Определение 2.5. *Пусть a и b – действительные числа, которые имеют одинаковый знак и не равны нулю, и q – целое, такие что $2^q \le |a|, |b| < 2^{q+1}$. Расстоянием между мантиссами a и b мы будем называть величину*

$$\frac{|a - b|}{2^q}.$$

Теорема 2.2 (В. Лефевр, Ж.-М. Мюллер [7]). *Пусть x – число двойной точности и y – его экспонента. Пусть, далее, y^* – приближенное значение y такое, что расстояние между мантиссами y и y^* ограничено ε. Тогда в следующих случаях*

- *если $|x| \ge 2^{-30}$ и $\varepsilon \le 2^{-113}$,*
- *если $2^{-54} \le |x| < 2^{-30}$ и $\varepsilon \le 2^{-158}$*

округление y^* эквивалентно округлению y для любого из рассматриваемых режимов округления.

Приведем две модификации теоремы 2.2. Первая относится ко всем рассматриваемым режимам округления.

Теорема 2.3. *Пусть x – число двойной точности и y – его экспонента. Пусть, далее, y^* – приближенное значение y, такое что расстояние между мантиссами y и y^* ограничено ε. Тогда в следующих случаях*

- $|x| \ge 2^{-37}$ *и* $\varepsilon \le 1.33 \cdot 2^{-113}$,
- $2^{-44} \le |x| < 2^{-37}$ *и* $\varepsilon \le 1.33 \cdot 2^{-134}$,
- $2^{-49} \le |x| < 2^{-44}$ *и* $\varepsilon \le 1.33 \cdot 2^{-149}$,
- $2^{-54} \le |x| < 2^{-49}$ *и* $\varepsilon \le 1.33 \cdot 2^{-158}$

округление y^* эквивалентно округлению y для любого из рассматриваемых режимов округления.

Доказательство следует из данных, приведенных в приложении А работы [4]. Вторая модификация относится к режимам округления к ближайшему (см. [4], теорема 6.1).

Теорема 2.4. *Пусть x – число двойной точности такое, что $|x| \ge 2^{-54}$, а y – его экспонента. Пусть, далее, y^* – приближенное значение y, такое что расстояние между мантиссами y и y^* ограничено $1.67 \cdot 2^{-112}$. Тогда для режимов округления к ближайшему округленное значение y^* равно округленному значению y.*

3. ОСОБЫЕ СЛУЧАИ

Положим

$$x_{ovr} = \texttt{0x1.62e42fefa39efp+9} \approx 709.78,$$
$$x_{dnrm} = -\texttt{0x1.6232bdd7abcd2p+9} \approx -708.40,$$
$$x_{zero1} = -\texttt{0x1.74385446d71c3p+9} \approx -744.44,$$
$$x_{zero2} = -\texttt{0x1.74910d52d3051p+9} \approx -745.13.$$

Пусть x – число двойной точности. С помощью непосредственных вычислений было получено следующее. Если $x > x_{ovr}$, то корректно округленное значение e^x равно $+\infty$ при всех видах округления, кроме округления к $-\infty$. При округлении к $-\infty$ оно равно $(2 - 2^{-52}) \cdot 2^{1023}$.

Если $x_{dnrm} \le x \le x_{ovr}$, то при всех видах округления корректно округленное значение e^x есть нормализованное число двойной точности.

Если $x < x_{dnrm}$, то при всех видах округления корректно округленное значение e^x есть ненормализованное число или 0.

Если $x < x_{zero1}$, то $e^x < 2^{-1074}$ (наименьшее положительное ненормализованное число). Поэтому корректно округленное значение e^x равно 0 в режиме округления к $-\infty$. В случае округления к $+\infty$ оно равно 2^{-1074}.

Если $x < x_{zero2}$, то $e^x < 2^{-1075}$. Поэтому в режиме округления к ближайшему и к $-\infty$ корректно округленное значение e^x равно 0. В случае округления к $+\infty$ оно равно 2^{-1074}.

4. АЛГОРИТМ

Предлагаемый алгоритм вычисления корректно округленного значения экспоненты состоит из нескольких (под)алгоритмов, которые реализуют стадии сокращения аргумента, аппроксимации и восстановления. Рассматриваемые алгоритмы используют при вычислениях арифметику расширенной двойной точности, а в результате их применения мы получаем корректно округленное значение экспоненты двойной точности.

4.1. Сокращение аргумента

Предполагается, что предварительно отфильтровываются нечисловые, а также слишком большие и слишком малые значения аргумента, которые обрабатываются отдельно. Детальное описание этого шага можно найти в [5]. Далее мы предполагаем, что аргумент принадлежит отрезку $[x_{zero2}, x_{ovr}]$.

Стадия сокращения аргумента состоит из четырех шагов (алгоритмы 4.1, 4.2, 4.3 и 4.4). Алгоритм 4.1 для сокращения аргумента использует равенство

$$e^x = 2^n \cdot e^{x - n\cdot \ln 2}.$$

Величину $\ln 2$ мы аппроксимируем тройкой $l = (l_1, l_2, l_3) \in T_{-1,-40,-77,-130}$. Путем непосредственных вычислений получаем

$$|l_1| < 0.694, \quad |l_2| < 1.517 \cdot 2^{-43},$$
$$|l_3| < 1.851 \cdot 2^{-79}, \qquad (4.1)$$
$$|\ln 2 - (l_1 + l_2 + l_3)| < 1.901 \cdot 2^{-137}.$$

Константу $1/\ln 2$ аппроксимируем числом расширенной двойной точности $\tilde{l}$:

$$\left|\frac{1}{\ln 2} - \tilde{l}\right| \le 2^{-64}. \qquad (4.2)$$

Алгоритм 4.1. Сокращение аргумента. Шаг 1

1: $m_0 \leftarrow R_0(x \cdot \tilde{l}\,)$
2: $x_{0,1} \leftarrow x - m_0 \cdot l_1 \triangleright x_{0,1}$ вычисляется точно, $|x_{0,1}| < 0.5 \cdot \ln 2 + 1.596 \cdot 2^{-33}$
3: $x_{0,2} \leftarrow -m_0 \cdot l_2 \quad \triangleright x_{0,2}$ вычисляется точно
4: $x_{0,3} \leftarrow -m_0 \cdot l_3 \quad \triangleright x_{0,3}$ вычисляется точно

Теорема 4.1. *Пусть x – плавающее число двойной точности такое, что $x_{zero2} \le x \le x_{ovr}$, числа m_0, $x_{0,1}$, $x_{0,2}$ и $x_{0,3}$ получены с помощью алгоритма* 4.1. *Тогда m_0 – целое число, $x_{0,1}$ и $x_{0,2}$ – плавающие числа двойной точности, а $x_{0,3}$ – плавающее число расширенной двойной точности, и справедливы следующие равенства и неравенства*:

$$x = m_0 \cdot \ln 2 + (x_{0,1} + x_{0,2} + x_{0,3}) - \delta_0,$$

$$\text{где} \quad |m_0| < 1.051 \cdot 2^{10}, \quad |\delta_0| < 1.998 \cdot 2^{-127},$$

$$|x_{0,1}| < 0.5 \cdot \ln 2 + 1.596 \cdot 2^{-33},$$

$$|x_{0,2}| < 1.595 \cdot 2^{-33}, \quad x_{0,2} \in B_{-33,-77},$$

$$|x_{0,3}| < 1.946 \cdot 2^{-69}, \quad x_{0,3} \in B_{-69,-130}.$$

В силу ограничений по объему статьи мы не приводим здесь доказательства теорем или ограничиваемся только основными идеями доказательств. Полные доказательства будут опубликованы позже.

Следующие три шага сокращения аргумента используют равенство

$$e^x = (1 + v) \cdot e^{x - \ln(1+v)}.$$

При этом мы выбираем значение v так, чтобы величина $x - \ln(1 + v)$ была по возможности малой, а мантисса числа v была короткой.

На втором шаге для поиска величин $1 + v$ и $\ln(1 + v)$ мы используем таблицу W_1, которая создается заранее следующим образом. Для каждого n в диапазоне от -89 до 89 мы находим ближайшее к $n \cdot 2^{-8}$ число вида $\ln(1 + m \cdot 2^{-8})$. Элемент номер n таблицы W_1 содержит плавающее число

$$W_1[n].w_0 = 1 + m \cdot 2^{-8} = 1 + v,$$
$$\text{где} \quad v = m \cdot 2^{-8}$$

и пару чисел расширенной двойной точности, аппроксимирующие $\ln(1 + m \cdot 2^{-8})$:

$$|\ln(1 + m \cdot 2^{-8}) - (W_1[n].w_1 + W_1[n].w_2)| < 2^{-129}.$$

Алгоритм 4.2. Сокращение аргумента. Шаг 2

1: $n_1 \leftarrow R_0(x_{0,1} \cdot 2^8)$

2: $M_1 \leftarrow W_1[n_1].w_0$

3: $x_{1,1} \leftarrow x_{0,1} - W_1[n_1].w_1$ ▷ вычисляется точно, $|x_{1,1}| < 1.175 \cdot 2^{-8}$

4: $x_{1,2} \leftarrow -W_1[n_1].w_2$

В начале второго шага мы находим ближайшее к $x_{0,1}$ число вида $n \cdot 2^{-8}$. Далее с помощью таблицы W_1 мы находим ближайшую к $n \cdot 2^{-8}$ величину вида $\ln(1 + m \cdot 2^{-8})$. Более точно, мы находим пару чисел расширенной двойной точности, аппроксимирующие $\ln(1 + m \cdot 2^{-8})$.

Теорема 4.2. *Пусть x – плавающее число двойной точности такое, что $x_{zero2} \le x \le x_{ovr}$, числа m_0, $x_{0,1}$, $x_{0,2}$ и $x_{0,3}$ получены с помощью алгоритма* 4.1, *а числа M_1, $x_{1,1}$ и $x_{1,2}$ получены с помощью алгоритма* 4.2.

Тогда $x_{1,1}$ и $x_{1,2}$ – плавающие числа расширенной двойной точности, и справедливы следующие равенства и неравенства:

$$x = m_0 \cdot \ln 2 + \ln(M_1) + x_{1,1} + x_{0,2} + x_{1,2} + x_{0,3}$$

$$-\delta_0 - \delta_1, \quad \text{где} \quad |\delta_0| < 1.998 \cdot 2^{-127}, \quad |\delta_1| \lesssim 2^{-129},$$

$$M_1 = 1 + v_1, \quad v_1 = m_1 \cdot 2^{-8}, \quad -75 \le m_1 \le 106,$$

$$|x_{1,1}| < 1.175 \cdot 2^{-8}, \quad x_{1,1} \in B_{-8,-64}, \quad \textit{если} \quad x \ge 2^{-12},$$

$$|x_{1,2}| \le 2^{-65}, \quad x_{1,2} \in B_{-65,-128}.$$

Третий шаг аналогичен второму. На этом шаге мы используем таблицу W_2, содержащую элементы с номерами от -75 до 75. Эта таблица позволяет нам аппроксимировать числа вида $n \cdot 2^{-14}$ числами вида $\ln(1 + m \cdot 2^{-14})$.

Алгоритм 4.3. Сокращение аргумента. Шаг 3

1: $n_2 \leftarrow R_0(x_{1,1} \cdot 2^{14})$

2: $M_2 \leftarrow M_1 \cdot W_2[n_2].w_0$ ▷ вычисляется точно, $0.703 < M_2 < 1.421$, $M_2 \in B_{0,-22}$

3: $x_{2,1} \leftarrow x_{1,1} - W_2[n_2].w_1$ ▷ вычисляется точно, $|x_{2,1}| < 0.675 \cdot 2^{-14}$

4: $x_{2,2} \leftarrow -W_2[n_2].w_2$

Теорема 4.3. *Пусть x – плавающее число двойной точности такое, что $x_{zero2} \le x \le x_{ovr}$, числа m_0, M_2, $x_{0,2}$, $x_{0,3}$, $x_{1,2}$, $x_{2,1}$ и $x_{2,2}$ получены путем последовательного применения алгоритмов* 4.1, 4.2 *и* 4.3. *Тогда $x_{2,1}$ – плавающее число двойной точности, $x_{2,2}$ – плавающее число расширенной двойной точности, и справедливы следующие равенства и неравенства*:

$$x = m_0 \cdot \ln 2 + \ln(M_2) + (x_{2,1} + x_{0,2}) + (x_{1,2} + x_{2,2} + x_{0,3}) - (\delta_0 + \delta_1 + \delta_2), \quad \textit{где}$$

$$M_2 = M_1 \cdot (1 + v_2), \quad v_2 = m_2 \cdot 2^{-14},$$

$$-75 \le m_2 \le 75,$$

$$0.703 < M_2 < 1.421, \quad M_2 \in B_{0,-22},$$

$$|x_{2,1}| < 0.675 \cdot 2^{-14},$$

$$x_{2,1} \in B_{-15,-67}, \quad \textit{если} \quad x \ge 2^{-15},$$

$$|x_{2,2}| \le 2^{-65}, \quad x_{2,2} \in B_{-65,-128}, \quad |\delta_2| \le 2^{-129}.$$

На четвертом шаге таблицы не применяются, а величины v и $\ln(1+v)$ вычисляются. Для поиска величины v мы используем приближение $x \approx \ln(1 + x)$. Положим $v_3 = 3 \cdot R_{-30}\left(\frac{1}{3} \cdot x_2\right)$, где $x_2 = x_{2,1} + x_{0,2}$. Величина v_3 имеет короткую мантиссу, делится точно на 3 и $x_2 \approx \ln(1 + v_3)$. Для вычисления $\ln(1 + v_3)$ мы используем следующее приближение

$$\ln(1 + v) \approx v - 0.5 \cdot v^2 + \frac{v^3}{3} + v^4 \cdot P_3(v). \qquad (4.3)$$

Полином P_3 был найден с помощью пакета Sollya [6]:

$$P_3(v) = a_0 + v \cdot (a_1 + v \cdot (a_2 + v \cdot a_3)),$$

$$a_0 = \texttt{-0xf.ffffffffffffdp-6},$$

$$a_1 = \texttt{0xc.ccccccccccccc1p-6},$$

$$a_2 = \texttt{-0xa.aaaaaae49a3c83p-6},$$

$$a_3 = \texttt{0x9.249249728e270fdp-6}.$$

Погрешность приближения (4.3) на интервале $[-0.677 \cdot 2^{-14}, 0.677 \cdot 2^{-14}]$ не превышает $1.495 \cdot 2^{-123}$.

Используя (4.3), получаем

$$(x_{2,1} + x_{0,2}) + (x_{0,3} + x_{1,2} + x_{2,2}) = \ln(1 + v_3) + ((x_{2,1} + x_{0,2}) + (x_{0,3} + x_{1,2} + x_{2,2}) - \ln(1 + v_3)) \approx \ln(1 + v_3) + x_{3,1} + x_{3,2}, \quad \text{где}$$

$$x_{3,1} = (x_2 - v_3) + 0.5 \cdot v_3^2 - v_3^2 \cdot v_3',$$

$$x_{3,2} = ((x_{0,3} + x_{1,2}) + x_{2,2}) - v_3^4 \cdot P_3(v_3).$$

Алгоритм 4.4. Сокращение аргумента. Шаг 4

1: $x_2 \leftarrow x_{2,1} + x_{0,2}$ $\triangleright$ x_2 вычисляется точно

2: $v_3' \leftarrow R_{-30}\left(\frac{1}{3} \cdot x_2\right)$

3: $v_3 \leftarrow 3 \cdot v_3' \triangleright |v_3| < 0.677 \cdot 2^{-14}$, $v_3 \in B_{-15,-30}$

4: $x_{3,1} \leftarrow (x_2 - v_3) + 0.5\, v_3^2 - v_3^2 \cdot v_3'$ $\triangleright$ $x_{3,1}$ вычисляется точно, $|x_{3,1}| < 1.210 \cdot 2^{-29}$

5: $x_{3,2} \leftarrow ((x_{0,3} + x_{1,2}) + x_{2,2}) - v_3^4 \cdot P_3(v_3)$

6: $\triangleright$ $|x_{3,2}| < 1.821 \cdot 2^{-61}$, $\Delta(x_{3,2}) < 1.219 \cdot 2^{-123}$

7: $M_3 \leftarrow M_2\,(1 + v_3)$ $\triangleright$ M_3 вычисляется точно, $0.703 < M_3 < 1.421$, $M_3 \in B_{0,-52}$

Теорема 4.4. *Пусть x – плавающее число двойной точности такое, что $x_{zero2} \le x \le x_{ovr}$, числа m_0, M_3, $x_{3,1}$ и $x_{3,2}$ получены путем последовательного применения алгоритмов* 4.1, 4.2, 4.3 *и* 4.4. *Тогда справедливы следующие равенства и неравенства*:

$$x = m_0 \cdot \ln 2 + \ln(M_3) + x_{3,1} + x_{3,2} - \delta_a,$$
$$\textit{где} \quad 0.703 < M_3 < 1.421, \quad M_3 \in B_{0,-52},$$
$$|x_{3,1}| < 1.210 \cdot 2^{-29}, \quad |x_{3,2}| < 1.821 \cdot 2^{-61},$$
$$|\delta_a| < 1.436 \cdot 2^{-122}.$$

Из теоремы 4.4 следует, что

$$e^x = 2^{m_0} \cdot M_3 \cdot e^{x_{3,1} + x_{3,2} - \delta_a}. \qquad (4.4)$$

Таким образом, в результате четырех шагов нам удалось сократить аргумент до величины меньшей $1.211 \cdot 2^{-29}$. Так как числа v_1, v_2 и v_3 имеют короткие мантиссы, то M_3 вычисляется точно и является числом двойной точности.

4.2. Вычисление значения функции

Алгоритмы 4.5 и 4.6 завершают вычисление корректно округленного значения экспоненты e^x. Если $x_{dnrm} \le x \le x_{ovr}$, то используется алгоритм 4.5, а если $x_{zero2} \le x < x_{dnrm}$, то используется алгоритм 4.6.

Фактически каждый из этих алгоритмов описывает три алгоритма: один – для округления к ближайшему, второй – для округления к плюс бесконечности и третий – для округления к минус бесконечности (см. комментарии к алгоритмам). Например, в алгоритме 4.5 при округлении к ближайшему используется строка 21, а строки с 22 по 39 не используются. Сами алгоритмы используют при вычислениях только округление к ближайшему (roundTiesToEven).

Теорема 4.5. *Пусть x – число двойной точности. Если $x_{dnrm} \le x \le x_{ovr}$, то алгоритмы* 4.1, 4.2, 4.3, 4.4 *и* 4.5 *вычисляют корректно округленное значение экспоненты e^x. Если $x_{zero2} \le x < x_{dnrm}$, то алгоритмы* 4.1, 4.2, 4.3, 4.4 *и* 4.6 *вычисляют корректно округленное значение экспоненты e^x.*

Полное доказательство теоремы 4.5 будет опубликовано позже.

5. ТЕСТИРОВАНИЕ И ВРЕМЕННЫЕ ХАРАКТЕРИСТИКИ

Предложенные выше алгоритмы были реализованы в виде функции crexp_e() на языке Си. Эта функция была протестирована на компьютере с процессором Core i7-2600K CPU, 3.4 ГГц и операционной системой Linux Fedora 20 Kernel 3.19.8. Для компиляции использовался компилятор gcc, версия 4.8.3, уровень оптимизации 3.

Для тестирования на каждом интервале $[x_{dnrm}, x_{ovr}]$ и $[x_{zero2}, x_{dnrm})$ случайным образом было выбрано по 10,000,000,000 значений аргумента, для которых вычислялось значение функции `crexp_e()` при всех рассматриваемых режимах округления. Дополнительно функция тестировалась при малых значениях аргумента ($|x| < 2^{-28}$) и при значениях аргумента близких к x_{dnrm}. Для тестирования использовались также худшие случаи из [2–4, 7]. Для проверки корректности тестируемой функции использовались функции библиотеки CRlibm, версия 1.0beta4. Все тесты прошли успешно.

При оценке производительности время измерялось в тактах процессора. Реальная частота процессора при измерениях была равна номинальной – 3.4 ГГц. Все измеряемые функции работали в режиме округления к ближайшему.

Мы также рассматривали экспоненциальную функцию ОС Линукс. Более точно, мы использовали функцию `__ieee754_exp()` вместо `exp()`, чтобы избежать динамического вызова. Эта функция вычисляет корректно округленное значение экспоненты при округлении к ближайшему. При этом используется только арифметика двойной точности.

Мы также рассматривали функцию Лаутера [8], которая вычисляет корректно округленное значение экспоненциальной функции при всех рассматриваемых режимах округления. Функция Лаутера использует арифметику расширенной двойной точности, написана на ассемблере и оптимизирована для процессора Intel Itanium.

Алгоритм 4.5. $x_{dnrm} \le x \le x_{ovr}$

1: $\triangleright\ e^x = 2^{m_0} \cdot M_3 \cdot e^{x_{3,1}+x_{3,2}-\delta_a}$
2: $\triangleright\ 0.703 < M_3 < 1.421,\ M_3 \in B_{0,-52}$
3: $\triangleright\ |x_{3,1}| < 1.210 \cdot 2^{-29},\ |x_{3,2}| < 1.821 \cdot 2^{-61},\ |\delta_a| < 1.436 \cdot 2^{-122}$
4: $M_{3,1} \leftarrow R_{-30}(M_3)$
5: $M_{3,2} \leftarrow M_3 - M_{3,1} \triangleright\ M_3 = M_{3,1} + M_{3,2}$
6: $S_1 \leftarrow R_{-62}(x_{3,1}) \triangleright\ S_1$ вычисляется точно, $|S_1| < 1.211 \cdot 2^{-29},\ S_1 \in B_{-29,-62}$
7: $x_3 \leftarrow x_{3,1} + x_{3,2}$
8: $\triangleright\ |x_3| < 1.211 \cdot 2^{-29},\ |\Delta(x_3)| \le 2^{-93}$
9: $S_2 \leftarrow ((x_{3,1} - S_1) + x_{3,2}) + x_3^2 \cdot \left(0.5 + \frac{1}{3!} \cdot x_3\right)$
10: $\triangleright\ |S_2| < 1.985 \cdot 2^{-59},\ |\Delta(S_2)| < 1.850 \cdot 2^{-121}$
11: $\triangleright\ e^{x_{3,1}+x_{3,2}-\delta_a} \approx 1 + S_1 + S_2$
12: $\triangleright\ 2^{-m_0} \cdot e^x \approx (M_{3,1} + M_{3,2}) \cdot (1 + S_1 + S_2) = M_3 + M_{3,1} \cdot S_1 + M_{3,2} \cdot S_1 + M_3 \cdot S_2$
13: $\triangleright\ |M_{3,1} \cdot S_1| < 1.723 \cdot 2^{-29},\ M_{3,1} \cdot S_1 \in B_{-29,-92}$
14: $\triangleright\ (2_{11} - 1) + M_3 \in B_{11,-52}$
15: $(Y, Y_1) \leftarrow Fast2Sum(((2^{11} - 1) + M_3), M_{3,1} \cdot S_1)$
16: $\triangleright\ |M_{3,2} \cdot S_1 + M_3 \cdot S_2| < 1.714 \cdot 2^{-58},\ |\Delta(M_{3,2} \cdot S_1 + M_3 \cdot S_2)| < 1.815 \cdot 2^{-120}.$
17: $Y_2 \leftarrow Y_1 + (M_{3,2} \cdot S_1 + M_3 \cdot S_2)$
18: $\triangleright\ |Y_2| < 1.054 \cdot 2^{-53},\ |\Delta(Y_2)| < 1.227 \cdot 2^{-117}$
19: $\triangleright$ Округленные значения e^x и $2^{m_0} \cdot ((Y - (2^{11} - 1)) + Y_2)$ равны
20: $\triangleright$ Округление к ближайшему
21: $y \leftarrow (Y + Y_2) - (2^{11} - 1)$ $\triangleright\ 0.701 + (2^{11} - 1) < Y + Y_2 < 1.423 + (2^{11} - 1)$
22: $\triangleright$ Округление к $+\infty$
23: $y \leftarrow Y - (2^{11} - 1)$ $\triangleright\ 0.702 \le y \le 1.422$
24: **if** $Y_2 > \mathbf{0}$ then
25: **if** $y \ge 1$ **then**
26: $y \leftarrow y + 2^{-52}$
27: **else**
28: $y \leftarrow y + 2^{-53}$
29: **end if**
30: **end if**
31: $\triangleright$ Округление к $-\infty$ или к 0
32: $y \leftarrow Y - (2^{11} - 1)$
33: **if** $Y_2 < 0$ **then**
34: **if** $y > 1$ **then**
35: $y \leftarrow y - 2^{-52}$
36: **else**
37: $y \leftarrow y - 2^{-53}$
38: **end if**
39: **end if**
40: **return** $y \cdot 2^{m_0}$ $\triangleright$ Восстановление

Алгоритм 4.6. $x_{zero2} \le x < x_{dnrm}$

1: $\triangleright\ e^x = 2^{m_0} \cdot M_3 \cdot e^{x_{3,1}+x_{3,2}-\delta_a}$
2: $\triangleright\ 0.703 < M_3 < 1.421,\ M_3 \in B_{0,-52}$
3: $\triangleright\ |x_{3,1}| < 1.210 \cdot 2^{-29}, x_{3,1} \in B_{-29,-90}$, $|x_{3,2}| < 1.821 \cdot 2^{-61}, |\delta_a| < 1.436 \cdot 2^{-122}$
4: $M_{3,1} \leftarrow R_{-30}(M_3)$
5: $\triangleright\ 0.702 < M_{3,1} < 1.422,\ M_{3,1} \in B_{0,-30}$
6: $M_3' \leftarrow 2^{m_0} \cdot M_3$
7: $M_{3,1}' \leftarrow 2^{m_0} \cdot M_{3,1}$
8: $M_{3,2}' \leftarrow M_3' - M_{3,1}'$
9: $\triangleright\ M_3' = M_{3,1}' + M_{3,2}'$
10: $S_1 \leftarrow R_{-62}(x_{3,1}) \triangleright S_1$ вычисляется точно, $|S_1| < 1.211 \cdot 2^{-29},\ S_1 \in B_{-29,-62}$
11: $x_3 \leftarrow x_{3,1} + x_{3,2}$
12: $\triangleright\ |x_3| < 1.211 \cdot 2^{-29},\ |\Delta(x_3)| \le 2^{-93}$
13: $S_2 \leftarrow ((x_{3,1} - S_1) + x_{3,2}) + x_2^3 \cdot \left(0.5 \cdot \frac{x_3}{3!}\right)$
14: $\triangleright\ |S_2| < 1.985 \cdot 2^{-59},\ |\Delta(S_2)| < 1.850 \cdot 2^{-121}$
15: $Y_1 \leftarrow t_2 + (t_1 + (M_{3,2} \cdot S_1 + M_3 \cdot S_2))$
16: $\triangleright\ e^x \approx M_3' \cdot (1 + S_1 + S_2)$
17: $\triangleright\ 2^{-1011} + e^x \approx 2^{-1011} + M_3' + M_{3,1}' \cdot S_1 + M_{3,2}' \cdot S_1 + M_3' \cdot S_2$
18: $(t_0, t_1) \leftarrow Fast2Sum(M_3', M_{3,1}' \cdot S_1)$
19: $\triangleright\ M_3', M_{3,1}' \cdot S_1, t_0$ и t_1 вычисляются точно
20: $(Y, t_2) \leftarrow Fast2Sum(2^{-1011}, t_0)$
21: $\triangleright\ Y \in B_{-1011,-1074}, Y$ и t_2 вычисляются точно
22: $\triangleright\ e^x \approx (Y - 2^{-1011}) + (t_2 + (t_1 + (M_{3,2}' \cdot S_1 + M_3' \cdot S_2)))$
23: $\triangleright$ Округление
24: $y \leftarrow Y - (2^{-1011} - 2^{-1022})$
25: $\triangleright$ Округление к ближайшему
26: **if** $(t_2 + 2^{-1075}) + (t_1 + (M_{3,2}' \cdot S_1 + M_3' \cdot S_2)) < 0$ **then**
27: $\quad y \leftarrow y - 2^{-1074}$
28: **else if** $(t_2 - 2^{-1075}) + (t_1 + (M_{3,2}' \cdot S_1 + M_3' \cdot S_2)) > 0$ **then**
29: $\quad y \leftarrow y + 2^{-1074}$
30: **end if**
31: $\triangleright$ Округление к $+\infty$
32: **if** $t_2 + (t_1 + (M_{3,2}' \cdot S_1 + M_3' \cdot S_2)) > 0$ **then**
33: $\quad y \leftarrow y + 2^{-1074}$
34: **end if**
35: $\triangleright$ Округление к $-\infty$
36: **if** $t_2 + (t_1 + (M_{3,2}' \cdot S_1 + M_3' \cdot S_2)) < 0$ **then**
37: $\quad y \leftarrow y - 2^{-1074}$
38: **end if**
39: return $y - 2^{-1022}$ $\triangleright$ Восстановление

Таблица 1. Максимальное время (нсек)

	$[x_{dnrm}, x_{ovr}]$	$[x_{zero2}, x_{dnrm})$
crexp_e	141	156
crexp_d	219	220
Lauter	172	4578
CRlibm	735	1023
Linux	76297	6160
Combined	185	157

Таблица 2. Среднее время (нсек)

	$[x_{dnrm}, x_{ovr}]$	$[x_{zero2}, x_{dnrm})$
crexp_e	141	156
crexp_d	188	195
Lauter	172	4578
CRlibm	85	1020
Linux	52	381
Combined	51	157

Для сравнения мы также использовали функцию `crexp_d()`, которая реализует алгоритмы предложенные автором в [4]. Эта функция использует только арифметику двойной точности.

Таблицы 1 и 2 содержат результаты измерений (в тактах процессора). Максимальное время выполнения функций на указанных интервалах приведено в таблице 1. Таблица 2 содержит среднее время выполнения функций. Все измерения для всех функций кроме функции Лаутера были произведены автором. Данные, относящиеся к функции Лаутера, взяты из статьи Лаутера [8].

Как видно из таблицы 1, функция `crexp_e()` обладает наименьшим максимальным временем выполнения на обоих интервалах. Функция Лаутера требует на 22% больше времени на интервале $[x_{dnrm}, x_{ovr}]$ и в 30 раз больше времени на интервале $[x_{zero2}, x_{dnrm})$.

Среднее время выполнения функции `crexp_e()` на интервале $[x_{zero2}, x_{dnrm})$ также меньше, чем у других функций, представленных в таблице, но среднее время выполнения функции `crexp_e()` на интервале $[x_{dnrm}, x_{ovr}]$ больше, чем у функций CRlibm и Линукс. Среднее время можно снизить объединяя функции `crexp_e()` и `__ieee754_exp()` аналогично тому, как это сделано в [4]. Для этого нужно в функции `__ieee754_exp()` заменить вызов функции `__slowexp()` на вызов функции `crexp_e()` на интервале $[x_{dnrm}, x_{ovr}]$. При этом на интервале $[x_{zero2}, x_{dnrm})$ вызывается только функция `crexp_e()`. Полученная таким образом функция

(Combined) имеет лучшее среднее время выполнения на обоих интервалах $[x_{zero2}, x_{dnrm})$ и $[x_{dnrm}, x_{ovr}]$, но максимальное время на интервале $[x_{dnrm}, x_{ovr}]$ увеличится. Отметим, что эта функция может использоваться только в режиме округления к ближайшему.

ИСТОЧНИК ФИНАНСИРОВАНИЯ

Публикация выполнена в рамках государственного задания по проведению фундаментальных исследований по теме “Исследование и реализация программной платформы для перспективных многоядерных процессоров” (FNEF-2022-002).

СПИСОК ЛИТЕРАТУРЫ

1. IEEE Std. 754-2008 – IEEE Standard for Floating-Point Arithmetic. IEEE Std. 2018.
2. *Lefèvre V., Muller J.-M.* Worst cases for correct rounding of the elementary functions in double precision. In Proceedings of the 15th IEEE Symposium on Computer Arithmetic. June 2001. P. 111–118.
3. *Lefèvre V.* Hardest-to-Round Cases – Part 2 // Journées TaMaDi. Lyon. Oct. 2013. [Online]. Available: http://tamadiwiki.enslyon.fr/tamadiwiki/images/c/c1/Lefevre2013.pdf.
4. *Godunov A.* Algorithms for Calculating Correctly Rounded Exponential Function in Double-Precision Arithmetic // IEEE Transactions on Computers. V. 69. № 5. P. 1–12. July 2020. https://doi.org/10.1109/TC.2020.2972901
5. *Daramy-Loirat C., Defour D., de Dinechin F., Gallet M., Gast N., Lauter C.Q., Muller J.-M.* “CR-LIBM, a library of correctly rounded elementary functions in double-precision”, LIP, Research Report, 2006, https://hal-enslyon. archives-ouvertes.fr/ensl-01529804.
6. *Chevillard S., Joldeş M., Lauter C.* Sollya: An Environment for the Development of Numerical Codes. In Mathematical Software – ICMS 2010. P. 28–31, Heidelberg, Germany, September 2010, Springer.
7. *Muller J.-M.* Elementary Functions: Algorithms and Implementation. Birkhauser. 2005.
8. *Lauter C.* A correctly rounded implementation of the exponential function on the Intel Itanium architecture // INRIA, Research Report, RR-5024, 2003. [Online]. Available: https://hal.inria.fr/inria-00071560/document

EDN: DMRAIX
ПРОГРАММИРОВАНИЕ, 2022, № 6, с. 22–30

АНАЛИЗ ДАННЫХ

УДК 004.032.26

ОПТИМИЗАЦИЯ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ С ПОМОЩЬЮ ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЙ

© 2022 г. Н. А. Вершков[a,*] (ORCID: 0000-0001-5756-7612),
М. Г. Бабенко[a,b,] (ORCID: 0000-0001-7066-0061),**
А. Н. Черных[b,c,d,*] (ORCID: 0000-0001-5029-5212),**
В. А. Кучуков[a,**] (ORCID: 0000-0002-1839-2765), Н. Н. Кучеров[a,*****] (ORCID: 0000-0003-0337-0093),**
Н. Н. Кучукова[a,****] (ORCID: 0000-0002-8070-0829)**

[a] Северо-Кавказский центр математических исследований,
Северо-Кавказский федеральный университет,
355017, г. Ставрополь, ул. Пушкина, 1, Россия

[b] Институт системного программирования РАН,
109004, г. Москва, ул. А. Солженицына, д. 25, Россия

[c] Центр научных исследований и высшего образования,
22860, Нижняя Калифорния, Энсенада, ш. Тихуана-Энсенада, 3918, Мексика

[d] Южно-Уральский государственный университет,
454080, Челябинск, проспект Ленина, 76, Россия

**E-mail: nvershkov@ncfu.ru*
***E-mail: mgbabenko@ncfu.ru*
****E-mail: chernykh@cicese.mx*
*****E-mail: vkuchukov@ncfu.ru*
******E-mail: nkucherov@ncfu.ru*
*******E-mail: nkuchukova@ncfu.ru*

Поступила в редакцию 01.07.2022 г.
После доработки 16.08.2022 г.
Принята к публикации 22.09.2022 г.

В статье рассматривается вопрос оптимизации искусственных нейронных сетей с точки зрения быстродействия с помощью вейвлет-преобразования. Существующие подходы внедрения вейвлет-преобразования в нейронные сети подразумевают или преобразование до нейронной сети, или с использованием архитектуры “вейвнета”, требующей новых подходов к обучению и применению нейронной сети. Предлагаемый подход опирается на представление модели нейрона в виде нерекурсивного адаптивного фильтра и применение вейвлет-фильтра для получения низкочастотной части изображения, снижая тем самым размер изображения и отфильтровывая помехи, имеющие, как правило, высокочастотную природу. Предлагаемая модель вейвлет-преобразования опирается на классическое представление слоя нейронной сети прямого распространения или сверточного слоя. Предлагаемый подход позволяет проектировать нейронные сети с вейвлет-преобразованием на базе существующих библиотек и не требует вносить изменения в алгоритм обучения нейронных сетей. Для подтверждения теоретических положений предлагаемый алгоритм был проверен на трех MNIST-подобных сетах. Для программистов-практиков предлагаемый алгоритм был проверен на реальных изображениях для различения животных и показал аналогичные результаты, что и на MNIST-подобных тестах.

DOI: 10.31857/S0132347422060073

1. ВВЕДЕНИЕ

Одной из самых широко распространенных задач, решаемых искусственными нейронными сетями (ИНС), является распознавание визуальных образов. ИНС способны распознавать рукописные и печатные символы на бумаге, подписи на официальных документах, водяные знаки и пр. подобные объекты. Применение ИНС позволяет существенно облегчить труд человека, повысить надежность и точность бизнес-процессов за счет снижения человеческого фактора. В общем виде распознавание образов – это отнесение визуального объекта к одному из классов (задача классификации), когда классы заранее определены, или выявление классов по некоторым общим призна-

кам и отнесение объекта к одному из них (задача кластеризации).

Вопросам решения задачи распознавания образов уделяется много внимания и посвящено большое количество литературы [1–4]. Одной из основополагающих работ, посвященных этой теме, считается книга Ахмеда Н. и Рао К.Р. [5] о роли ортогональных преобразований в цифровой обработке сигналов, в т.ч. при распознавании образов. Книга стала обобщением целого ряда работ авторов, посвященных обобщенным ортогональным преобразованиям, функциям Уолша, преобразованиям Хаара, а также методам вычисления спектра в различных базисах. Несмотря на значительный срок, работы Ахмеда Н. и Рао К.Р. до сих пор актуальны и находят применение в современных исследованиях.

Использование ИНС для распознавания образов, несмотря на значительное количество исследований, достаточно ограничено. Большинство авторов рассматривают ИНС в виде обучаемого классификатора, предпочитая формировать вектор признаков и осуществлять отбор признаков стандартными методами [6–9]. Действительно, классическая схема распознавания образов, предложенная в [5], включает в себя ортогональное преобразование, формирующее вектор признаков, алгоритм оптимизации вектора признаков с целью уменьшения размерности и обучаемый классификатор, чаще всего представляющий собой ИНС прямого распространения. Дело в том, что теория ИНС с самого зарождения опирается на представление многомерной нелинейной функции в виде суперпозиции одномерных функций, аргументами которых являются аргументы исходной функции или их комбинация [10–12]. Кроме того, расширение теоремы Колмогорова–Арнольда работами Хехт-Нильсена [13] ограничивает размер скрытого слоя ИНС снизу. В связи с этим постулатом формирования вектора признаков и его оптимизация в ИНС невозможны.

В то же время целый ряд работ [14–16] позволяют исследовать ИНС как информационную систему. Использование математической модели нейрона [17] как адаптивного сумматора (или нерекурсивного адаптивного фильтра) позволяют применить основные положения теории информации к теории ИНС [18]. В частности, в работе [19] авторы рассмотрели возможность применения пространственно-темпоральной модели нейрона для реализации возможности обработки нестационарных процессов на ИНС. Рассматриваются также возможности использования преобразования Фурье для фильтрации периодических помех, возникающих на космических снимках [9].

Попытки построения ортогональных нейронных сетей предпринимаются и в настоящее время, однако, как правило, носят узкоспециализированный характер, например, в медицине [20].

Основным недостатком преобразования Фурье является его низкая информативность во временной области. Обработка нестационарных сигналов требует различные степени разрешения как во временной, так и в частотной областях. Использование вейвлетов позволяет преодолеть ограниченность преобразования Фурье во временной области. Попытки создать гибридную нейронную сеть с использованием вейвлет-преобразования предпринимались многократно для различных направлений. Так в работе [21] вейвлет-преобразование применялось для предварительной обработки временных рядов и были получены результаты, значительно улучшающие качество работы ИНС по сравнению с классической моделью. Однако, в работе не описан метод выполнения преобразования. В работе [22] авторы исследовали возможности по использованию вейвлет-преобразования для распознавания символов. Опираясь на опубликованные результаты, авторы утверждают, что использование вейвлет-преобразования позволяет сократить время обучения ИНС минимум в 3 раза, что, несомненно, является значимым результатом. Причины успеха, которые называют авторы, связаны с физической природой преобразования Хаара и со снижением размерности вектора признаков. К сожалению, в работе не приведена архитектура ИНС, поэтому сложно судить о глубине применения ИНС для решения задачи преобразования Хаара.

Подводя итог вышеприведенному обзору необходимо отметить, что процесс использования ИНС для классификации и кластеризации изображений требует решения 2-х основных задач: максимальная унификация ИНС, т.е. интеграция процессов нахождения вектора признаков в слои ИНС и оптимизация ИНС, т.е. снижения ее размерности, путем оптимизации вектора признаков. При этом решение задачи оптимизации должно происходить с учетом первой задачи, т.е. в рамках архитектуры ИНС.

Появление сверточных ИНС [23, 25, 26] в корне изменило подход к решению задачи распознавания. Резкое снижение количества обучаемых параметров ИНС позволило значительно повысить качество обучения: так для датасета MNIST [32] за менее, чем 20 эпох достигается качество 98% для сверточного слоя 1d. Для слоя 2d скорость и качество обучения еще выше. Сверточная архитектура ИНС включает в себя 3 основных парадигмы: локальное восприятие, разделяемые веса и субдискретизация. За счет этого происходит снижение размерности ИНС, однако, как правило, последние слои ИНС, предназначенные для классификации признаков, остаются слоями прямого распространения и количество связей в

них значительно. Поэтому количество признаков остается критическим параметром.

В работе [18] авторы предложили для формирования вектора признаков дискретное преобразование Фурье (ДПФ) в слое ИНС прямого распространения. Предлагаемый подход опирается на представление ДПФ в виде фильтра. Такое представление позволило использовать модель нейрона МакКаллока–Питтса [17] для формирования частотной характеристики в каждом нейроне первого слоя. При этом слой выступает как линейный частотный преобразователь. Использование вместо ДПФ таких ортогональных преобразований как, например, дискретное косинусное преобразование (ДКП) позволяет снизить количество нейронов в последующих слоях ИНС. Однако отбор значимых признаков осуществлялся с помощью дисперсионного критерия и требовал отдельного исследования в процессе обучения.

Использование Фурье-подобных преобразований осложняется еще тем, что частотное представление сигнала не всегда позволяет выделить лучший вектор признаков по сравнению с временным представлением. Например, когда входной сигнал ИНС не является стационарным (в этом случае шум при представлении одного кластера не является гауссовым), тогда частотный спектр также не будет стационарным в силу влияния каждого временного отсчета на каждую частотную составляющую.

Если же вместо Фурье-подобных преобразований использовать вейвлет-преобразование, то можно получить выигрыш в количестве значимых признаков без проведения дополнительных исследований. Вейвлет-преобразование представляет на выходе аппроксимацию, т.е. низкочастотную часть сигнала и отдельно – высокочастотную. Поскольку шум обычно имеет высокочастотную природу, то данный подход может быть использован для очищения сигналов от шума путем отбрасывания высокочастотной составляющей входного сигнала, выделенной путем вейвлет-преобразования. Применение вейвлетов в ИНС на сегодняшний день не является новостью [24, 25], появился даже термин “вейвнет”, означающий нейроподобную среду с использованием вейвлетов в качестве активационных функций. Однако подобный подход имеет 2 момента, усложняющих обучение ИНС: во-первых, в процессе обучения изменяются не только веса нейронов, но и параметры вейвлетов, а во-вторых, ИНС начинает принимать несвойственную ей пирамидальную архитектуру, которая также скажется на алгоритме обучения. И, наконец, подобный подход к использованию вейвлетов в ИНС потребует создания собственных библиотек поддержки архитектуры “вейвнетов”, что ставит под сомнение целесообразность данного подхода.

Поэтому авторы в рамках данной статьи поставили себе задачу унификации и оптимизации ИНС при использовании вейвлетов, а также необходимость оставаться в рамках популярных библиотек для построения вейвлет-ориентированных ИНС.

Статья построена следующим образом: во втором разделе рассматривается вейвлет-преобразование как способ снижения размера изображения и формирования вектора признаков оптимального размера и максимальной информативности. В третьем разделе проводится сравнительная характеристика ИНС, не использующей вейвлет-преобразования, с ИНС, которая реализует в первом слое вейвлет-преобразование. При этом анализируется не только быстродействие, но и качество обучения ИНС. В заключение сформулированы результаты применения вейвлет-преобразования, а также определены направления дальнейших исследований.

2. ИСПОЛЬЗОВАНИЕ ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЙ ДЛЯ СЖАТИЯ ВХОДНОГО СИГНАЛА ИНС

Как отмечалось ранее во введении, преобразование Фурье, несмотря на его очень широкое применение, обладает целым рядом недостатков при анализе нестационарных сигналов. К этим недостаткам относятся, в первую очередь, низкая пространственная локализация, т.к. спектр сигнала формируется с учетом всех значений сигнала:

$$S(\omega) = \sum_n s_n(t)\, u_n(t) \qquad (1)$$

Ортогональность базисных функций имеет большое значение [26–28]: однозначность значений коэффициентов разложения по базису существует только в случае, если

$$\int u_m(t)\, u_n(t)\, dt = \begin{cases} c, & \text{если } m = n \\ 0, & \text{если } m \neq n \end{cases}$$

В случае, когда $c = 1$ базисные функции ортонормированы. Для улучшения временной локализации используют дополнительную функцию $g(t - t_0)$, которая позволит преобразовать (1) к виду:

$$S(\omega) = \sum_n s_n(t)\, g_n(t - t_0)\, u_n(t)$$

Такое преобразование называют оконным преобразованием Фурье. Оконное преобразование частично решает вопрос временной локализации, однако обладает собственными недостатками, к важнейшему из которых относят принцип неопределенности Гейзенберга. В этом

случае говорят не о конкретной частоте, а о диапазоне частот, присутствующих в сигнале в определенном диапазоне времени.

Вейвлет-анализ имеет много общего с классическим ортогональным преобразованием [24, 25], фактически является разновидностью ортогональных преобразований. Как, например, преобразование Фурье в качестве базовой функции использует комплексную экспоненту $e^{in\omega_0 t}$, где $i = \sqrt{-1}$, n – номер гармоники, ω_0 – основная (центральная) частота, t – время, так и вейвлет-преобразование имеет базовую функцию. Например, широко известный на практике вейвлет Хаара описывается функцией вида

$$\psi(t) = \begin{cases} +1, & 0 \le t < 0.5 \\ -1, & 0.5 \le t < 1 \\ 0, & t < 0, \quad t \ge 1 \end{cases} \qquad (2)$$

Рассматривая преобразование Хаара, необходимо отметить, что половина коэффициентов преобразования Хаара соответствуют корреляции соседних точек в пространстве входных последовательностей, четверть – связям четырех соседних точек и т.д. В отличии от ДПФ и преобразования Уолша–Адамара преобразование Хаара обладает свойством как локальной, так и глобальной чувствительности, благодаря чему именно преобразование Хаара стало первым вейвлет-преобразованием, применяемым на практике.

Базис пространства $L^2(R)$ конструируют из порождающей функции $\psi(t)$, например (2), с помощью двух преобразований независимой переменной: $\psi(t) \Rightarrow \psi(t+k)$ и $\psi(t) \Rightarrow \psi(a^m t)$. Первое преобразование реализует сдвиг порождающей функции по оси времени для покрытия ограниченной функцией $\psi(t)$ (например, функцией Хаара) всей области преобразования, а второе (масштабирование) – для формирования покрытия спектра сигнала. Объединив оба преобразования в одно выражение, получим

$$\psi(t) \Rightarrow \psi(a^m t + k) \qquad (3)$$

Подобный подход в теории связи называют частотно-временной матрицей, применяют при синтезе сложных сигналов, он хорошо описан в [30]. Таким образом, произвольная функция в гильбертовом пространстве может быть разложена по базису $\psi_{m,k}(t)$ и представлена в виде ряда

$$s(t) = \sum_{m,k} S_{mk} \psi_{mk}(t) \qquad (4)$$

Спектр вейвлет-преобразования является двумерным в пространстве переменных m и k в соответствии с (3). Однако, и одномерный спектр, построенный для определенного значения m, дает информацию, которая может быть использована для анализа. Так, коэффициент масштабирования a^m влияет на ширину функции по оси времени, а значит, и на полосу частот, занимаемую этой функцией. Небольшие значения коэффициента масштабирования соответствуют узким базисам во временной области и, соответственно, более высоким частотам и, наоборот, рост коэффициента масштабирования приводит к более “длинным” базисам и низким частотам. Таким образом, каждая строка спектра, проведенная параллельно оси времени, дает представление об активности исследуемого процесса в определенной полосе частот на определенном промежутке времени и, следовательно, обладает чувствительностью как во временной, так и в частотной области. Рассмотрим теперь подробнее реализацию вейвлет-преобразования в ИНС как прямого распространения, так и сверточных.

Поскольку вейвлет-преобразование является разновидностью ортогональных преобразований и определяется выражением (4), то по аналогии с [18] для ИНС прямого распространения можно воспользоваться моделью МакКаллока–Питтса [17]:

$$y_{k,l} = f\left(\sum_{i=1}^{n} w_i^{k,l} x_i^{k,l}\right), \qquad (5)$$

где l, k – номер слоя и номер нейрона в слое, $y_{k,l}$ – выход нейрона, $x_i^{k,l}$ – входы нейрона, $w_i^{k,l}$ – веса (синапсы) входных сигналов, f – выходная функция нейрона. Сравнив выражения (1) и (5) можно убедиться в том, что значение под функцией полностью идентично одной строке спектра вейвлета, а, следовательно, может быть получен набор коэффициентов вейвлет-преобразования в первом приближении. Для этого необходимо использовать слой нейронов, в котором в качестве весов занесены значения вейвлет-функции $\psi(a^m t + k l_\psi)$, где $a, m = \text{const}$, k – номер нейрона в слое, а l_ψ – длина вейвлета. Все остальные значения весов должны быть обнулены, а также должен быть введен запрет на изменение весов в этом слое. Этот способ апробирован при вычислении ДПФ, ДКП и т.п. [18].

Предложенный выше способ дает стабильный результат, но обладает значительным недостатком: для получения одной строки вейвлет-спектра используется полноценный слой с размерностью равной размерности сигнала (более подробно это будет показано в следующем разделе). При этом большая часть весов будет обнулена, т.к. размерность вейвлета по сравнению с размерностью входного сигнала обычно невелика. Можно, конечно, отказаться от нейронов, у которых веса обнулены, но в этом случае придется “расплетать” входной сигнал, т.е. перенаправлять соот-

ветствующие разряды входного сигнала на соответствующие входы нейронов.

Для снижения количества используемых связей можно использовать сверточный слой. Процедура вейвлет-преобразования может быть описана как операция пропускания входного сигнала через полуполосный цифровой фильтр с частотной характеристикой $h(n)$ (высокочастотный фильтр) или $g(n)$ (низкочастотный фильтр):

$$\begin{cases} x(n) * h(n) = \sum_k x(k)\, h(n-k) \\ x(n) * g(n) = \sum_k x(k)\, g(n-k) \end{cases} \quad (6)$$

Если сигнал на входе ИНС представляет собой одномерную последовательность длиной n, то, используя одномерный сверточный слой с ядром $h(n)$ или $g(n)$, на выходе получим коэффициенты вейвлет-преобразования. Чтобы сократить количество слоев ИНС, можно использовать один слой с двумя (или более) различными ядрами. Для этого создается сверточный слой с одним входом, двумя выходами и с шагом, равным размерности вейвлета. В этом случае будет создан сверточный слой с двумя ядрами, в которые заносятся значения $h(n)$ и $g(n)$.

Для реализации многомерных вейвлет-преобразований придется задействовать количество слоев, равное количеству измерений. На примере двумерного вейвлет-преобразования это понять несложно. Пусть входное изображение представлено матрицей пикселов $l \times l$. На первом этапе осуществляется построчное вейвлет-преобразование, например, вида (6). Затем осуществляется транспонирование полученного результата и повторное вейвлет-преобразование. К сожалению, объединить эти этапы не представляется возможным, т.к. ко второму этапу преобразования можно перейти только после полного выполнения вейвлет-преобразования всех строк. При увеличении размерности входного сигнала пропорционально растет и количество слоев вейвлет-преобразования.

Таким образом, используя слои прямого распространения или сверточные слои, предоставляемые стандартными библиотеками, например, Keras, PyTorch и т.п., можно осуществлять выполнение операции вейвлет-преобразования. 1d вейвлет-преобразование фактически разделяет спектр сигнала на 2 части: низкочастотную и высокочастотную. Для целей классификации (кластеризации) будем использовать низкочастотную часть, т.к. основная часть шума имеет высокочастотную природу и, как показывает практика, не оказывает значительного влияния на качество классификации. Хотя, при необходимости, могут быть задействованы и обе составляющие [21] или высокочастотная часть, например, для определения границ.

3. ПРАКТИЧЕСКИЕ АСПЕКТЫ ПРИМЕНЕНИЯ ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЯ В ИНС

Для практической оценки возможности оптимизации ИНС с помощью вейвлет-преобразования была использована несложная сверточная сеть с двумя сверточными слоями и двумя слоями прямого распространения для реализации обучаемого классификатора [31]. ИНС реализована с помощью стандартной библиотеки PyTorch [36]. В качестве данных для обучения ИНС и контроля обучения использовались 3 базы данных: MNIST [32], KMNIST [33] и Fashion-MNIST [34]. MNIST – база образцов рукописного написания арабских цифр. MNIST является стандартом, предложенным Национальным институтом стандартов и технологий США с целью калибровки и сопоставления методов распознавания изображений с помощью машинного обучения в первую очередь на основе ИНС. KMNIST (Kuzushiji-MNIST) является заменой набора данных MNIST, представленного в оригинальном формате MNIST. Поскольку MNIST ограничена 10 классами, то выбран один символ для представления каждого из 10 рядов Хираганы при создании Kuzushiji-MNIST. KMNIST создан Центром открытых данных в гуманитарных науках ROIS-DS (CODH). Fashion-MNIST – это также альтернатива MNIST. Он создан немецкой компанией Zalando и охватывает в общей сложности 70 000 различных изображений из 10 категорий. Подразделение Fashion-MNIST по размеру, формату, комплекту обучения и тестирования точно такое же, как и исходный MNIST. Использование трех различных баз, использующих одинаковый размер и организацию данных, сформулировано несколько пренебрежительным отношением многих программистов-практиков к базе MNIST и желанием получить более надежные гарантии работоспособности предлагаемого метода. Применение же MNIST-подобных баз данных позволило получить сравнительные данные без изменения структуры ИНС, т.е. сохранены условия для качественного сравнения результатов тестирования.

Для оценки результатов обучения использовались результаты контрольных замеров на тестовых данных. Максимальное количество эпох, установленное для обучения, равно 20. В испытаниях принимали участие реализации вейвлет-преобразования с помощью линейных слоев ИНС и с помощью сверточных слоев.

Контрольные замеры были выполнены на всех 3 базах данных без использования вейвлет-преобразования, результаты представлены на рис. 1.

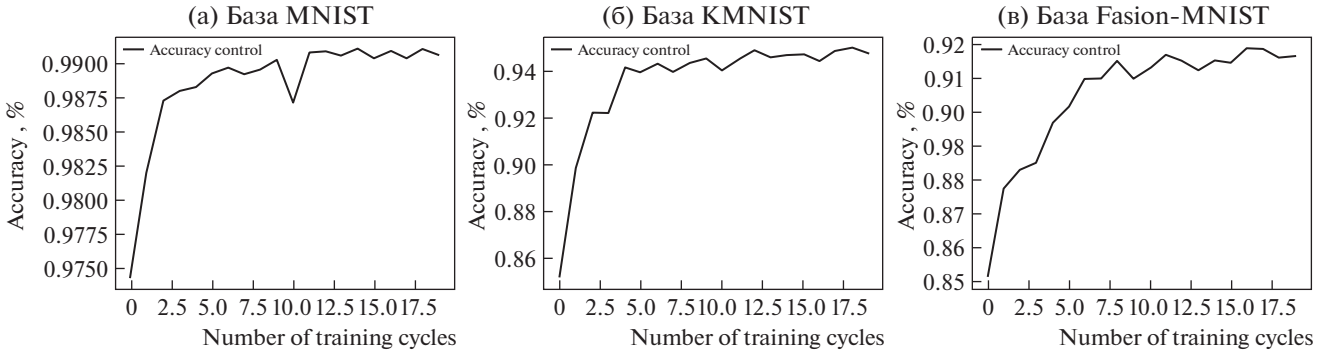

Рис. 1. Результаты обучения ИНС без вейвлет-преобразования.

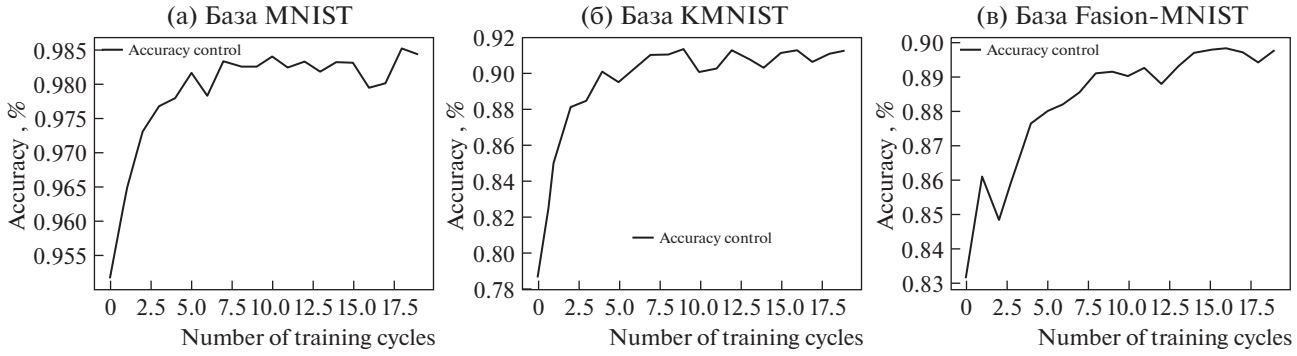

Рис. 2. Результаты обучения ИНС с 1 линейным слоем вейвлет-преобразования.

В результате обучения были получены следующие значения: обучение на базе MNIST дало максимальное значение качества обучения 99.1%, на базе KMNIST – 95.01%, на базе Fashion-MNIST – 91.87%. При этом производились замеры времени, которое было затрачено на 1 цикл обучения, данные об этих измерениях представлены в табл. 1.

Далее на вход ИНС был добавлен один линейный слой размером 784 нейрона (28 × 28 – размеры входного изображения). В качестве весов для каждого нейрона были установлены значения низкочастотного фильтра вейвлета Хаара (он же вейвлет Добеши 1 порядка). Причем в каждом следующем нейроне значения сдвигались на величину $s \times l$, где s – номер нейрона, а l – длина низкочастотного фильтра (для вейвлета Хаара равна 2). Результаты обучения с использованием линейного слоя вейвлет-преобразования представлены на рис. 2, а временные затраты – в таблице 2.

Получены следующие результаты обучения: для базы MNIST максимальное значение качества обучения составило 98.51%, на 0.59% ниже, чем без вейвлет-преобразования, для базы KMNIST – 91.37%, потеря качества обучения составила 3.64%, а для базы Fashion-MNIST – 89.81, потеря качества – 2.06%.

Если рассчитать в процентах, то выигрыш от использования вейвлет-преобразования по среднему времени обучения для базы MNIST составил 46.6% при потере качества обучения на 0.59%, для базы KMNIST – 54.4% при потере качества на 3.64%, для базы Fashion-MNIST – 50.6% при потере качества 2.06%. Таким образом, применение даже одного линейного низкочастотного фильтра вейвлет-преобразования позволяет получить зна-

Таблица 1. Затраты времени на 1 цикл обучения, мс

База	Макс. время	Мин. время	Среднее время
MNIST	0.2884	0.0758	0.1178
KMNIST	0.2015	0.0896	0.1286
Fashion-MNIST	0.3241	0.0948	0.1550

Таблица 2. Затраты времени на 1 цикл обучения с 1 линейным слоем вейвлет-преобразования, мс

База	Макс. время	Мин. время	Среднее время
MNIST	0.2067	0.0398	0.0629
KMNIST	0.1494	0.0369	0.0586
Fashion-MNIST	0.1254	0.0518	0.0766

Таблица 3. Затраты времени на 1 цикл обучения с 1 сверточным слоем вейвлет-преобразования, мс

База	Макс. время	Мин. время	Среднее время
MNIST	0.1498	0.0409	0.0617
KMNIST	0.1199	0.0389	0.0571
Fashion-MNIST	0.1167	0.0389	0.0597

чительный выигрыш (≈50%) в затратах времени при незначительной потере качества обучения.

Проведем сравнение характеристик исходной ИНС с нейронной сетью с вейвлет-преобразованием Хаара в первом сверточном слое. Для этого создадим сверточный слой $nn.Conv1d(1, 2, kernel\_size = dec\_len, stride = dec\_len, bias = False)$, где 1 – количество входов, 2 – количество выходов, *kernel_size* – размер ядра (для вейвлета Хаара 2), *stride* – шаг свертки, dec_len – длина фильтра вейвлета, полученная с помощью библиотеки PyWavelets, сверточный слой построен на базе библиотеки PyTorch [36] на основе модуля *nn*. В ядро преобразования занесем значение вейвлет-фильтра и также, как и для линейного слоя ранее, запретим изменение весов слоя. Результаты обучения приведены на рис. 3, а временные показатели – в таблице 3.

Сравнивая таблицы 2 и 3 нетрудно заметить, что результаты по затратам времени очень близки, небольшой выигрыш заметен только для базы Fashion-MNIST. Тем не менее, предпочтительнее использовать сверточный слой. Выигрыш заключается в том, что линейный слой выполняет только один вид вейвлет-преобразования – высокочастотное или низкочастотное. Для получения одновременно обоих результатов потребуется 2 линейных слоя ИНС. Сверточный слой формирует ядра таким образом, что преобразование от 1 входа к 2 выходам осуществляется обособленно двумя ядрами, куда и можно занести значения вейвлет-фильтра.

И, наконец, учитывая недоверчивое отношение программистов-практиков к экспериментам, выполненным с использованием баз данных MNIST, был проведен эксперимент по распознаванию вида животного по фотографии. Для этого был использован набор изображений собак и кошек с сайта Microsoft [35]. Для эксперимента была создана сверточная сеть на базе библиотеки PyTorch [36], содержащая 3 сверточных слоя и 2 слоя прямого распространения с двумя выходами. Результаты ее обучения с использованием вейвлета и без него на протяжении 10 эпох представлены на рис. 4, а затраты времени – в таблице 4.

Из рис. 4 видно, что модель ИНС с использованием вейвлета проигрывает в точности распознавания на 2.9% (76.1% против 79%), но выигрывает более, чем на 50% по среднему времени цикла обучения (табл. 4). Т.е. статистика, полученная

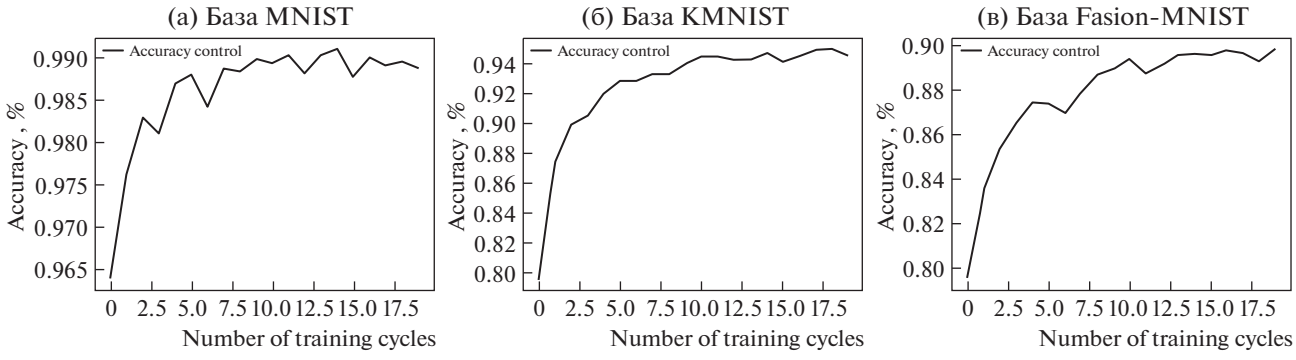

Рис. 3. Результаты обучения ИНС с 1 сверточным слоем вейвлет-преобразования.

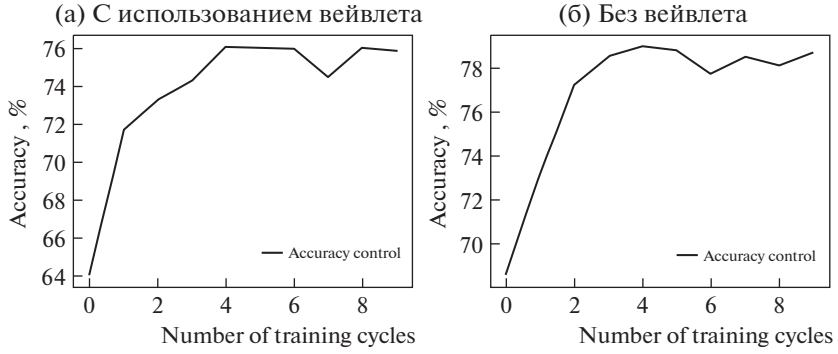

Рис. 4. Результаты обучения ИНС с 1 сверточным слоем вейвлет-преобразования.

на базах MNIST, подтверждается и на базах с реальными изображениями в тех же пропорциях: потеря 2-3% в точности распознавания при приблизительно двукратном выигрыше в скорости обучения.

Таким образом, вейвлет-преобразование в сверточном слое осуществляется одновременно и независимо, не требует формирования “вейвнетов” пирамидальной структуры. Кроме того, в качестве ядра преобразования может быть использован фильтр любого масштаба преобразования, а не только последовательно (например, в первом слое может быть использован фильтр Добеши сразу 2, 3 и более высокого порядка без использования первого). Следовательно, вейвлет-преобразование может быть интегрировано в архитектуру ИНС без дополнительных библиотек и доработок. Использование вейвлет-преобразования позволяет формировать удачный вектор признаков, чувствительный как к временным, так и частотным параметрам, а использование одной его части (низкочастотной) позволяет осуществлять оптимизацию объема вектора признаков (снижение в 2 раза) при незначительной потере качества распознавания и без необходимости проведения дополнительной оптимизации, как, например, с использованием дисперсионного критерия.

4. ЗАКЛЮЧЕНИЕ

Проведенные исследования показали эффективность использования вейвлет-преобразования для оптимизации работы ИНС. Вейвлет-преобразование гораздо более эффективно для формирования и снижения размерности вектора признаков, чем Фурье-подобные преобразования, т.к. обладает чувствительностью как во временной, так и в частотной области. Использование слоев ИНС, созданных стандартными библиотеками, для вейвлет-преобразования позволяет встроить процедуру формирования и оптимизации вектора признаков в архитектуру ИНС без создания дополнительных библиотек и проведения дополнительных исследований по отбору признаков. Результаты исследования, описанные в 3 разделе статьи показывают, что использование вейвлет-преобразования Хаара позволяет снизить временные затраты вдвое (пропорционально уменьшению размерности вектора признаков) при потере качества обучения 0.5–3%, что является хорошим показателем. Подобный подход, несомненно, будет востребован в тех случаях, когда необходимо осуществлять обработку больших объемов изображений (например, видео), а также при обработке изображений, имеющих априори высокую избыточность (высокое разрешение – HD, Full HD).

Для полноценного использования вейвлет-преобразований в ИНС необходимо провести еще ряд исследований. В первую очередь, это – построение полноценного двумерного спектра вейвлет-преобразования и его обработка средствами стандартных библиотек для проектирования ИНС. Это позволит полностью встроить процедуру вейвлет-преобразования в архитектуру ИНС и получить более широкий спектр возможностей по обработке и распознаванию изображений. Кроме того, значительный интерес представляет вейвлет-обработка цветных изображений, т.е. параллельная обработка 3-х слоев одновременно. Возможно, что здесь заложены значительные возможности по снижению временных затрат ИНС.

Таблица 4. Затраты времени на 1 цикл обучения со слоем вейвлет-преобразования и без него, мс

	Макс. время	Мин. время	Среднее время
С вейвлетом	0.3351	0.0558	0.1066
Без вейвлета	0.343	0.1047	0.2259

5. ФИНАНСИРОВАНИЕ

Работа выполнена при поддержке Российского научного фонда, проект № 19-71-10033.

СПИСОК ЛИТЕРАТУРЫ

1. Методы компьютерной обработки изображений / Под ред. В.А. Сойфера. 2-е изд., испр. М.: ФИЗМАТЛИТ, 2003 г.
2. Компьютерное зрение. Современный подход: Пер. с англ. М.: Издательский дом “Вильямс”, 2004 г.
3. *Шапиро Л., Стокман Дж.* Компьютерное зрение: Пер. с англ. М.: Бином. Лаборатория знаний, 2006 г.
4. *Hastie T., Tibshirani R., Friedman J.* The Elements of Statistical Learning: Data Mining, Inference, and Prediction. New York: Springer, 2009. Т. 2. С. 1–758.
5. *Ахмед Н., Рао К.Р.* Ортогональные преобразования при обработке цифровых сигналов: Пер. с англ. / Под ред. И.Б. Фоменко. М.: Связь, 1980 г.
6. *Ranzato M.A., Poultney C., Chopra S., Cun Y.* Efficient learning of sparse representations with an energy-based model // Advances in neural information processing systems. 2006. Т. 19.
7. *Галушкин А.И., Томашевич Д.С., Томашевич Н.С.* Методы реализации инвариантности к аффинным преобразованиям двумерных изображений // Приложение к журналу “Информационные технологии”. 2001. № 1. С. 1–19.
8. *Филлипс П.Дж., Мартин Э., Уилсон С.Л., Пржибоски М.* Введение в оценку биометрических систем // Открытые системы. 2000. № 3. С. 21–27.
9. *Гусев В.Ю., Крапивенко А.В.* Методика фильтрации периодических помех цифровых изображений // Труды МАИ. Радиотехника. Электроника. Телекоммуникационные системы. 2012. № 50.
10. *Колмогоров А.Н.* О представлении непрерывных функций нескольких переменных в виде суперпозиций непрерывных функций одного переменного и сложения. Докл. АН СССР. 1957. Т. 114. № 5. С. 953–956.
11. *Арнольд В.И.* О представлении функций нескольких переменных в виде суперпозиции функций меньшего числа переменных // Мат. просвещение. 1958. Вып. 3. С. 41–61.
12. *Горбань А.Н.* Обобщенная аппроксимационная теорема и вычислительные возможности нейронных сетей. Сибирский журнал вычислительной математики. 1998. Т. 1. № 1. С. 12–24.
13. *Hecht-Nielsen R.* Neurocomputing. Addison-Wesley, 1989. 433 p.
14. *Widrow B.* Adaptive sampled–data systems, a statistical theory of adaptation. IRE WESCON Convention Record, 1959. V. 4. P. 74–85.
15. *Цыпкин Я.З.* Информационная теория идентификации. М., Наука. Физматлит, 1995 г.
16. *Харкевич А.А.* Теория информации. Опознавание образов. Избранные труды в трех томах. Т. III. М., “Наука”, 1973 г.
17. *Мак-Каллок У., Питтс У.* Логическое исчисление идей, относящихся к нервной активности // Автоматы. М.: Изд. иностр. лит., 1956. С. 362–384.
18. *Vershkov N.A., Kuchukov V.A., Kuchukova N.N., Babenko M.* The Wave Model of Artificial Neural Network. In Proc. of the 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, EIConRus. 2020. P. 542–547.
19. *Малыхина Г.Ф., Меркушева А.В.* Метод контроля состояния подсистемы (объекта) при неполной измерительной информации о совокупности параметров, определяющих ее динамику. Научное приборостроение. 2004. Т. 14. № 1. С. 72–84.
20. *Kim J.S., Cho Y., Lim T.H.* Prediction of locations in medical images using orthogonal neural networks // European Journal of Radiology Open. 2021. Т. 8. С. 100388.
21. *Jamal A., Ashour M.A.H., Helmi R.A.A., Fong S.L.* A Wavelet-Neural Networks Model for Time Series // 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE). IEEE, 2021. С. 325–330. https://doi.org/10.1109/ISCAIE51753.2021.9431777
22. *Хаустов П.А., Григорьев Д.С., Спицын В.Г.* Разработка системы оптического распознавания символов на основе совместного применения вероятностной нейронной сети и вейвлет-преобразования // Известия Томского политехнического университета. Инжиниринг георесурсов. 2013. Т. 323. № 5. С. 101–105.
23. *LeCun Y., Bengio Y.* Convolutional networks for images, speech, and time series // The handbook of brain theory and neural networks. 1995. Т. 3361. № 10. С. 1995.
24. *Fujieda S., Takayama K., Hachisuka T.* Wavelet convolutional neural networks // arXiv preprint arXiv:1805.08620. 2018.
25. *Veitch D.* Wavelet Neural Networks and their application in the study of dynamical system. Departament of Mathematics University of York. 2005
26. *Нагорнов О.В., Никитаев В.Г., Простокишин В.М., Тюфлин С.А., Проничев А.Н., Бухарова С.А., Чистов К.С., Кашафутдинов Р.З., Хоркин В.А.* Вейвлет анализ в примерах. М.: НИЯУ МИФИ, 2010.
27. *Kerenidis I., Landman J., Mathur N.* Classical and quantum algorithms for orthogonal neural networks // arXiv preprint arXiv:2106.07198. 2021.
28. *Wang J., Chen Y., Chakraborty R., Yu S.X.* Orthogonal convolutional neural networks // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020. С. 11505–11515. https://doi.org/10.1109/CVPR42600.2020.01152
29. *Смоленцев Н.К.* Основы теории вейвлетов. Вейвлеты в MATLAB. М.: ДМК Пресс, 2019.
30. *Сикарев А.А., Лебедев О.Н.* Микроэлектронные устройства формирования и обработки сложных сигналов. М.: Изд-во “Радио и связь”, 1983
31. *Koehler G.* MNIST Handwritten Digit Recognition in PyTorch. Nextjournal, 2020, https://nextjournal.com/gkoehler/pytorch-mnist Accessed 28.02.2022.
32. Qiao, Yu THE MNIST DATABASE of handwritten digits (2007). Accessed 04.03.2022.
33. GitHub – rois-codh/kmnist: Repository for Kuzushiji-MNIST, Kuzushiji-49, and Kuzushiji-Kanji Accessed 04.03.2022.
34. A MNIST-like fashion product database. Benchmark https://github.com/zalandoresearch/fashion-mnist. Дата обращения 04 марта 2022.
35. Cats and Dogs Dataset https://download.microsoft.com/download/3/E/1/3E1C3F21-ECDB-4869-8368-6DEBA77B919F/kagglecatsanddogs_3367a.zip Дата обращения 04 марта 2022.
36. PyTorch (англ.) https://pytorch.org/get-started/previous-versions/ Дата обращения 04 марта 2022.

EDN: XSMGFA
ПРОГРАММИРОВАНИЕ, 2022, № 6, с. 31–45

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

УДК 004.85.056.5:004.93'1

СКРЫТЫЙ МОНИТОРИНГ ПОЛЬЗОВАТЕЛЯ В ДИСТАНЦИОННОЙ ОБРАЗОВАТЕЛЬНОЙ СИСТЕМЕ НА ОСНОВЕ КЛАВИАТУРНОЙ ДИНАМИКИ

© 2022 г. Е. А. Кочегурова[a,*] (ORCID: 0000-0003-4473-528X),
Р. П. Затеев[a,] (ORCID: 0000-0001-8852-0737)**

[a] Национальный исследовательский Томский политехнический университет
634050, г. Томск, пр. Ленина., д. 30, Россия
**E-mail kocheg@mail.ru*
***E-mail: roma-zateev@mail.ru*

Поступила в редакцию 22.06.2022 г.
После доработки 04.07.2022 г.
Принята к публикации 06.07.2022 г.

Пандемия ускорила развитие дистанционного обучения, важную роль в котором играют онлайн-тесты и экзамены. Особую актуальность при онлайн-тестировании имеет обнаружение подмены личности тестируемого и других фактов академического мошенничества. Способом противодействия несанкционированному доступу может быть непрерывная биометрическая (поведенческая) аутентификация. В работе предложена технология проверки легитимности тестируемого на основе его клавиатурной динамики в режиме скрытого мониторинга. Создано программное приложение для сбора, актуализации образцов клавиатурного почерка пользователей домена и непрерывной аутентификации личности. Показана эффективность сокращения размерности пространства клавиатурных признаков на основе частотности букв алфавита. Традиционные показатели эффективности (FAR, FRR, ERR, ROC, DET) заметно улучшены уже при оценке только метрических расстояний. Например, универсальная ошибка ERR снизилась с 10.1% до 0.79% и сопоставима с оценками kNN-метода для оптимальных значений параметров этого метода.

DOI: 10.31857/S0132347422060048

1. ВВЕДЕНИЕ

Пандемия COVID-19 оказала огромное влияние на функционирование и развитие практически всех сфер жизни. Для снижения инфицирования многие компании и организации ввели удаленный режим работы и другие ограничительные меры. И тем самым ускорили развитие процессов цифровизации, дистанционного обучения, телемедицины, интернет-торговли и других процессов.

Но положительный технологический толчок развития ИТ-отрасли привел к неизбежному росту киберпреступности. В год начала пандемии (2020) и в мире, и в России зафиксирован взрывной рост числа преступлений в сфере компьютерной информации. И согласно данным «АНО “Цифровая экономика” за 2020 год в России зафиксировано 363 тыс. киберпреступлений, что на 77% больше, чем за предыдущий год» [1]. Согласно статистическим данным группы компаний InfoWatch наряду с увеличением количества атак хакерских группировок, возникли новые риски, связанные с удаленной работой [2]. В 2021 г. в мире зафиксирован резкий рост умышленных утечек информации (82% от общего количества) и утечек от действий внешних киберпреступников (до 63%). А общая доля России во всемирном количестве утечек информации довольно внушительна и составляет 16.9%.

В 2021 году по статистике компании Ivideon число кибератак по сравнению с предыдущим годом выросло на 40% в мире и на 54% в России.

И хотя обеспечение информационной безопасности всегда являлось одной из ключевых задач любой организации, в нынешних условиях эта проблема особенно актуальна.

Сфера образования относится к наиболее уязвимым с точки зрения кибербезопасности – ввиду ментальности контингента обучаемых и наличия огромных массивов конфиденциальной информации. Это привлекает университеты для хакерских атак. Только за сентябрь 2021 г. произошло более 10 инцидентов с участием вирусов-вымогателей персональных данных о студентах и преподавателях с последующим предложением о выкупе [3].

С другой стороны, пандемия привела к критическому сбою функционирования традиционных систем университетского и школьного образования. И по данным ЮНЕСКО [4] “кризис затронул почти 1.6 миллиарда учащихся в более чем 190 странах на всех континентах. Закрытие школ и других образовательных учреждений коснулось 94% мирового контингента учащихся”. И хотя в сентябре 2021 года [5] число частично или полностью закрытых образовательных учреждений уменьшилось до 7.5%, но формы и способы подачи знаний значительно трансформировались в сторону дистанционных технологий. Однако готовность университетов к удаленному обучению на сегодня невысокая и функционально ограниченная. В университетском образовании online обучение зачастую сводится к вебинарам посредством видео/аудио конференций и работе на той или иной платформе электронного образования (Moodle, WebTutor, iSpring Learn и др.).

Онлайн-тесты и экзамены имеют большое значение в электронном обучении. Тесты позволяют преподавателю получать обратную связь от студента, оценивать его знания и совершенствовать обучение. Однако иногда студенты используют ряд методов академического мошенничества во время онлайн-тестов, включая выдачу себя за другое лицо [6, 7]. И на сегодня не существуют простых методов обнаружения таких подмен.

2. ОБ АКАДЕМИЧЕСКОЙ ЧЕСТНОСТИ ОНЛАЙН-ОБУЧЕНИЯ

Академическая честность – это проблема не только сферы образования, но и всего общества. Академические нарушения негативно влияют на качество образовательной среды, приобретенные студентами компетенции и общий имидж университета.

Повсеместный перевод вузовского образования в онлайн-формат неизбежно активизировал многие традиционные формы академического мошенничества: плагиат, фальсификация результатов, несанкционированное сотрудничество, подмена личности и пр. Кроме организационных средств, противодействия такому мошенничеству (университетский Кодекс поведения студентов, санкции администрации за академические нарушения), можно выделить методические рекомендации для преподавателей. Для онлайн-занятий рекомендуется:

– повысить индивидуальность и конкретность заданий;

– снизить фактологическую часть заданий и повысить концептуальную для развития мышления;

– увеличить число синхронных проверок ответов в онлайн-режиме;

– увеличить количество мелких и простых заданий, а для больших использовать свободный формат, например эссе.

Предложенные [8] меры частично снижают нарушения образовательной этики, но заметно увеличивают нагрузку на преподавателя.

Способом противодействия несанкционированному сотрудничеству и подмены личности может быть непрерывная аутентификация (НА) студента в скрытом режиме.

3. ВОПРОСЫ КЛАВИАТУРНОЙ АУТЕНТИФИКАЦИИ И ИДЕНТИФИКАЦИИ

Нередко для защиты компьютерной системы от несанкционированного доступа используется двухэтапный процесс верификации:

– первичная идентификация – установление личности пользователя, т.е. подтверждение легитимности авторизованного пользователя;

– динамическая (непрерывная) аутентификация, т.е. непрерывное подтверждение личности легитимного пользователя.

3.1. Методы аутентификации

Аутентификация – это процесс сравнения данных, представленных пользователем, с учетными данными, хранящимися в базе данных служб каталогов. При их совпадении пользователь получает доступ к защищенным ресурсам, при несовпадении – доступ запрещен [9].

Существует несколько методов аутентификации пользователя, рис. 1. Методы можно разделить на три основные категории, исходя из следующих парадигм [10]:

– что вы знаете (например, пароль, PIN-код). Аутентификация на основе знаний;

– чем вы владеете (например, токен, смарт-карта). Аутентификация на основе атрибутов

– кем вы являетесь. Физиологическая и поведенческая биометрия.

Парадигма “кем вы являетесь” связана с биометрическими признаками: физиологическими (отпечаток пальца, лицо, радужная оболочка глаза и пр.) и/или поведенческими (рукописный или клавиатурный почерк, походка, и др.) [19].

Распознавание пользователя на основе клавиатурной динамики довольно привлекательно для организации и имеет более низкую стоимость в сравнении с другими биометрическими методами, поскольку не требуется дополнительного оборудования. Кроме стандартной клавиатуры, требуется высокоэффективное программное приложение. Помимо точности, у этого метода

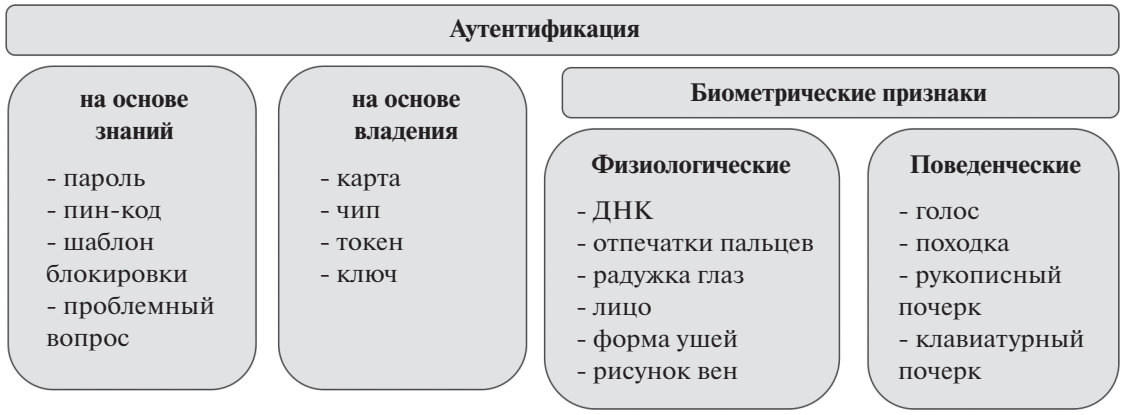

Рис. 1. Методы аутентификации личности.

есть преимущество в виде распознавания пользователей в фоновом, т.е. в скрытом и комфортном для человека режиме. Индивидуальные нажатия клавиш, ритм, скорость набора определяют клавиатурный почерк (КП) пользователя. Все поведенческие характеристики, в том числе и КП, могут постепенно меняться со временем, однако вероятность украсть или имитировать эти данные существенно ниже, чем у физиологических. Поэтому, КП можно использовать для идентификации и аутентификации пользователя [11, 12]. Преимущества, недостатки и примеры вышеописанных методов аутентификации кратко сведены в табл. 1.

3.2. Режимы аутентификации

Наиболее обоснованный для системы распознавания и комфортный для пользователя способ идентификации личности – это постоянный и скрытый мониторинг динамики его работы. И по данным глобального опроса по безопасности IBM (2018 г.) 44% респондентов считают биометрию самым безопасным методом аутентификации, а 65% обнаружили, что биометрия упрощает процесс аутентификации [13].

КП, как поведенческая биометрическая характеристика, является динамической. Она включает условно постоянную и случайную компоненты. Постоянная – обусловлена физиологией человека, его умением, способностями и навыком при работе на клавиатуре. Случайная компонента зависит от психоэмоционального состояния человека.

Для распознавания динамические поведенческие характеристики КП более сложны, чем физиологические. Но именно поведенческие характеристики сложны для подделки и подмены пользователя [10]. Что повышает эффективность для обнаружения самозванцев.

В зависимости от типа создаваемого текста, фиксированного или произвольного, можно выделить два основных режима аутентификации [14]:

– статическая (первичная или по событию);

– динамическая (непрерывная).

Именно НА позволяет организовать скрытую проверку личности пользователя в течение всего сеанса работы в любом программном приложении. Система распознавания фиксирует клавиатурные нажатия и сопоставляет их с уже имеющимся шаблоном пользователя [6, 11, 17–19].

Таблица 1. Характеристики методов идентификации

Метод	Достоинства	Недостатки	Пример
Парольный	1. Простая реализация 2. Однозначное распознавание	1. Может быть забыт или украден	1. Пароль 2. ПИН-код
Атрибутный	1. Простая реализация 2. Не требует затрат	1. Может быть потерян или украден	1. Ключ 2. Смарт-карта 3. Токен
Биометрический	1. Уникальность 2. Невозможно забыть/потерять	1. Стоимость реализации 2. Изменчивость данных независимо от человека	1. Отпечаток пальца 2. Голос 3. Клавиатурный почерк

Таблица 2. Исследования динамической идентификации

Год	Ссылка, автор	Параметр	Метод	Эффективность
2005	[25] Gunetti	FT	Расстояние, R/A	FAR-0.005%, FRR-5%
2010	[32] Shimshon		Кластеризация	FAR 3.47% FRR 0%
2011	[33] Messerman		Статистические	FAR-2.02%, FRR-1.84%
2011	[37] Solami		Кластеризация	Точность 100%
2013	[27] Alsultan	ди-граф	Смешанная (Fusion)	FAR-21%, FRR-17%
2014	[35] Ahmed	ди-граф	Нейронные сети	FAR-0.015%, FRR-4.82%
2014	[39] Antal	DT, FT	Статистические Метод опорных векторов Нейронные сети Дерево решений	Точность 93.04%
2014	[40] Locklear		Статистические	EER 4.55–13.37%
2015	[41] Kang	DT, FT	Кластеризация, Расстояние	EER 3.8%
2015	[42] Matsubara	ди-граф, DT	Расстояние	Точность 99%
2016	[23] Morales	ди-граф, n-граф	kNN, Расстояние	Точность 90%
2017	[31] Alsultan	ди-граф, DT	Метод опорных векторов	FAR 0.169, FRR 0.423
2017	[36] Goodkind	Contextual features	Наивный Байес	Точность 82.2%
2017	[30] Ali		kNN-метод	EER 3.7%
2021	[34] Chang	DT, FT	CNN-GRU	Точность 99% EER 0.0690

Статическая аутентификация может дополнять первичный вход в систему, либо активизируется при возникновении подозрений в злоумышленнике [15, 16].

Оба решения и статическая, и непрерывная аутентификации обеспечивают второй уровень защиты, когда пользователь уже находится в системе. Но при этом динамическая аутентификация нацелена на постоянную проверку легитимности пользователя или его психоэмоционального состояния.

Целью данного исследования является аутентификация личности пользователя на основе непрерывного мониторинга особенностей динамических характеристик его КП в условиях онлайн-тестирования. Для реализации данной цели работа сфокусирована на следующих задачах:

– сбор и актуализация динамических образцов КП для пользователей домена;

– расширение известных подходов статической идентификации пользователей для случая динамического распознавания личности на основе свободных и длинных текстов;

– снижения размерности пространства выделенных клавиатурных признаков для повышения селективных свойств образцов КП и эффективности НА.

3.3. Жизненный цикл аутентификации

НА пользователя на основе КП имеет фазу регистрации и фазу аутентификации, как показано на рис. 2.

На этапе регистрации система получает данные о клавиатурных нажатиях. Далее производится извлечение характеристик нажатий (длительность, паузы и пр.), формируется или модифицируется клавиатурный профиль (шаблон) в базе данных и производится аутентификация пользователя.

Т.о. жизненный цикл НА включает 4 основных этапа:

I. Сбор данных о динамике нажатия клавиш – это непрерывный процесс при работе пользователя на клавиатуре в любом программном приложении. Для ОС Windows задействован механизм пе-

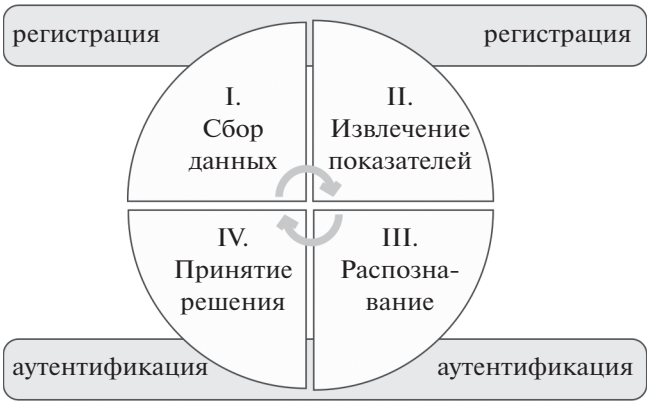

Рис. 2. Жизненный цикл непрерывной аутентификации.

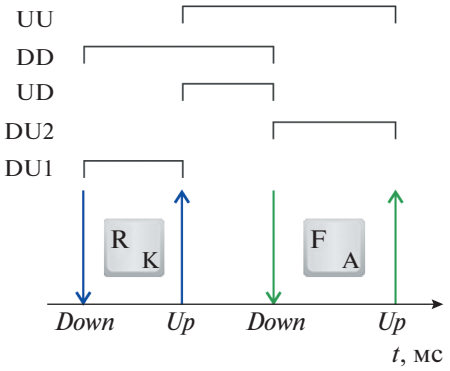

Рис. 3. Показатели нажатия клавиш в нотации Down Time/Up Time.

рехвата сообщений. С помощью Windows-hook можно зафиксировать любое событие использования клавиш [20]. ОС фиксирует код ANSI и временные метки нажатия клавиши (Down/Press) и ее отпускания (Up/Release). Точность измерения клавиатурных нажатий – миллисекунды.

II. Извлечение классификационных признаков Предварительно сырые данные о нажатиях клавиш должны быть очищены от выбросов, недостоверных значений и, в некоторых ситуациях, нормализованы. На основании этих данных можно получить ряд более значимых показателей КП: о ритме, скорости набора, паузах, отражающих уникальные поведенческие характеристики пользователя. Показателей КП довольно много, но наиболее популярны у исследователей ди-граммы (диграфы) – тайминг или временные метки двух состояний клавиши [17, 21–23]. Основные показатели следующие:

– время удержания клавиши;

– паузы между нажатиями;

– скорость набора;

– число ошибок при вводе;

– степень аритмичности при наборе;

– особенности использования служебных клавиш.

На рис. 3 приведены некоторые наиболее часто используемые временные и частотные показатели тайминга.

– DU – время удержания клавиши (ВУК) (Dwell Time, DT) – временной интервал между нажатием (Down, Press) и отпусканием (Up, Release) клавиши.

– UD – время между нажатиями или пауза (Flight Time, F-RP) – временной интервал между отпусканием пользователем предыдущей клавиши и нажатием следующей.

– UU, F-PP или DD, F-RR– интервал между нажатием или отпусканием одной клавиши и нажатием или отпусканием следующей клавиши соответственно

Подсистемы предварительной обработки информации о временных метках и извлечение показателей КП формируют массив требуемых показателей о каждом нажатии клавиши пользователем. Далее на основе этого массива генерируется клавиатурный профиль (шаблон) пользователя для размещения его в базе данных.

Банк профилей – это результат сбора характеристик КП, который требует адаптации. Как поведенческая биометрическая характеристика клавиатурный профиль изменчив и ему необходима актуализации. Коррекция клавиатурного профиля основана на технологии растущего или скользящего окна [49]. Используется банк профилей для обучения классификатора и на этапе распознавания личности пользователя

III. Распознавание пользователя

Нередко для защиты компьютерной системы от несанкционированного доступа используется двухэтапный процесс верификации:

– первичная идентификация, т.е. установление личности пользователя;

– динамическая (непрерывная) аутентификация, т.е. непрерывное подтверждение личности легитимного пользователя.

По типу задачи аутентификация – это задача классификации зарегистрированных в системе пользователей.

Основные методы и алгоритмы классификации (распознавания) пользователей одинаковы для статической и непрерывной (динамической) аутентификации. Их можно разделить на три группы:

– статистические;

– на основе оценки близости;

– методы машинного обучения.

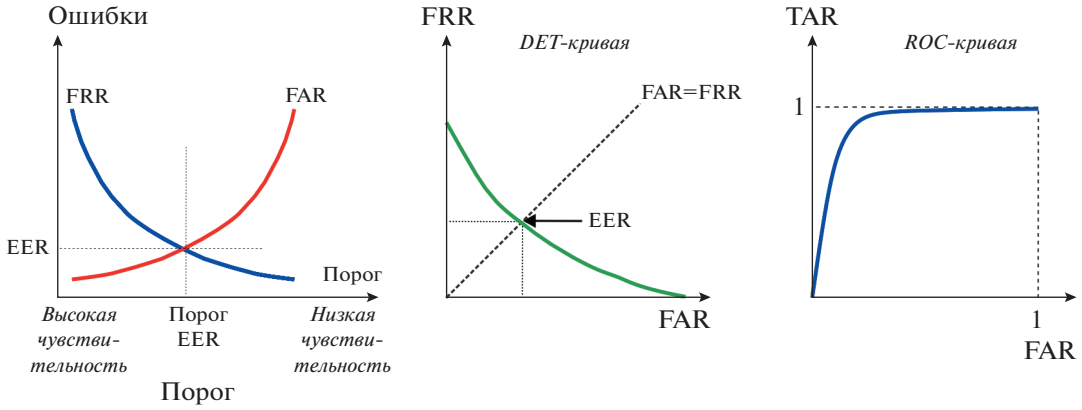

Рис. 4. Показатели эффективности клавиатурной идентификации.

Частота использования методов и основные представители в каждой группе подробно изложены в [20].

Исторически основные работы по распознаванию клавиатурной динамики относились к предопределенным и структурированным текстам, т.е. к статической аутентификации. И по данным разных авторов [22, 24] количество исследований, относящихся к НА по-прежнему невелико, и не превышает 10–15% от всего клавиатурного распознавания. Первым результативным исследованием НА можно считать работу Gunetti 2005 года, точность распознавания в которой превысила 90% [25]. Для контраста точность в самых первых результатах распознавания с использованием свободного текста (1997 г., Monrose) составляла 23%.

Обзорные работы по клавиатурному распознаванию последнего десятилетия позволили обобщить данные об эффективности НА, основные результаты приведены в табл. 2. Кроме библиографических ссылок таблица включает классификационный параметр, метод распознавания и показатели эффективности. Данные получены на основе собственных исследований [20, 26] и адаптированы из обзорных статей [17, 22, 24, 27–33].

В заключение анализа эффективности следует отметить условный характер подходов к распознаванию пользователей на основе его КП. Подходы, как правило, включают модель и методы обучения, но при этом их сочетания могут быть различными.

IV. Принятие решения о легитимности пользователя

Этот этап полностью определяется целями прикладной задачи на основании показателей эффективности распознавания.

При динамической идентификации основная цель непрерывного мониторинга – постоянный доступ к ресурсам сети для зарегистрированного в домене пользователя и предотвращение доступа незарегистрированному.

Следуя поставленной цели, принято оценивать вероятности соответствующих ложных событий: ложного отказа в доступе зарегистрированному и ложного доступа незарегистрированного пользователя. И по аналогии с оценками в статистической радиотехнике в исследованиях клавиатурной динамики чаще других используют следующие ошибки [17, 20, 21, 36, 38]:

– ошибка I рода False Rejection Rate (FRR) – частота ложного отказа в доступе законному (зарегистрированному) пользователю:

$$FRR = \frac{FR}{TA + FA + TR + FR}. \quad (1)$$

– ошибка II рода False Acceptance Rate (FAR) – частота ложного допуска к системе незаконных пользователей:

$$FRR = \frac{FA}{TA + FA + TR + FR}. \quad (2)$$

В (1) и (2) приняты обозначения:

– True Accept (TA) – число верных допусков в систему законным пользователям.

– True Reject (TR) – число верных отказов в доступе незаконным пользователям.

– False Accept (FA) – число ложных допусков незаконным пользователям.

– False Reject (FR) – число ложных отказов в доступе законным пользователям.

В знаменателях (1) и (2) – общее количество попыток.

Показатели FRR и FAR зависят от настраиваемого порога или чувствительности алгоритма, рис. 4.

Показатели FRR, FAR, ERR являются самодостаточными для принятия решения о допуске/отклонении пользователя. И, если цель системы мониторинга, высокая степень защиты, то следу-

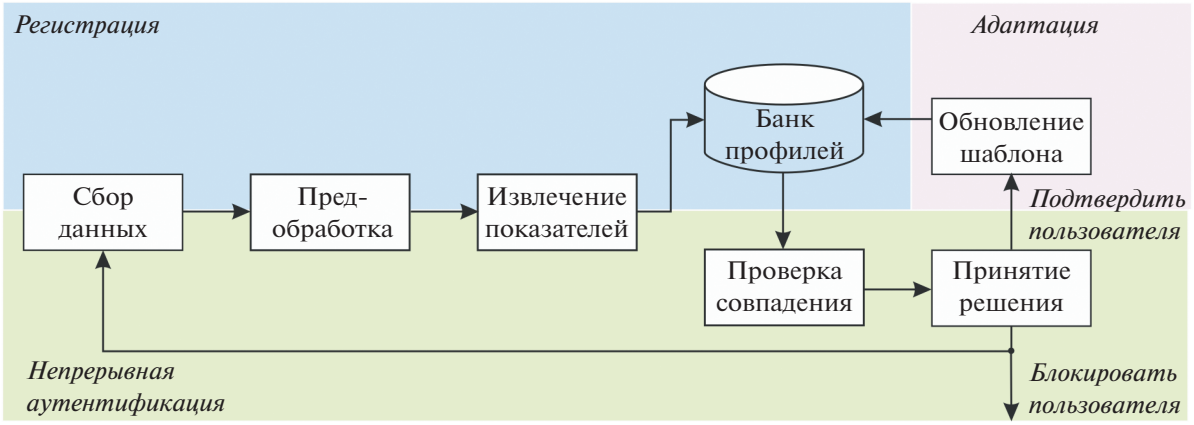

Рис. 5. Архитектура системы непрерывной аутентификации.

ет использовать низкие пороговые значения, которым соответствует большой процент ложно отклоненных (FRR). Большие значения FRR обеспечивают безопасность и более сложный вход в систему для всех – своих и чужих. А при высоком пороге и низкой чувствительности алгоритмов распознавания доступ упрощается, но вместе с тем возрастает FAR и число незаконно проникших пользователей. Вот этот компромисс между FRR и FAR приходится устанавливать индивидуально в каждой прикладной задаче.

Не менее популярным у исследователей является показатель, не зависящий от порогового значения – Equal Error Rate (EER). Значения EER соответствуют равным значениям FRR и FAR, что делает EER универсальным показателем для любой системы аутентификации.

Эти три показателя (FRR, FAR, EER) наиболее популярны при принятии решения в системах мониторинга и аутентификации. В научных исследованиях клавиатурной динамики также часто используется показатель Receiver Operating Characteristic (ROC) – соотношение между TA верно допущенными пользователями и FA ложно допущенными при различных пороговых значениях. ROC отражает предельные возможности алгоритмов, что особенно ценно при исследовании различных классификаторов.

4. СКРЫТЫЙ МОНИТОРИНГ В ДИСТАНЦИОННОЙ ОБРАЗОВАТЕЛЬНОЙ СИСТЕМЕ

Следует отметить, что пандемия ускорила внедрение модели смешанного обучения и многие университеты внедрили LMS – системы управления онлайн-обучением. Основной вклад в итоговую оценку студента в таких системах вносят онлайн-экзамены и тесты. Но именно эти формы онлайн-обучения более всего подвержены академическому мошенничеству, потому что онлайн-среда позволяет студентам работать практически без контроля. И, по мнению ряда исследователей [43–45], успех систем онлайн-экзаменов заключается в использовании биометрических систем контроля и непрерывного мониторинга во время онлайн-экзамена. Клавиатурная динамика позволяет организовать скрытый мониторинг личности легитимного студента (прошедшего первичную идентификацию) в комфортном для него режиме.

Основные вопросы клавиатурной динамики – жизненный цикл, компоненты, эффективность – рассмотрены в разделе 3. Особенности использования этих технологий приведены в последующих разделах 4.1–4.4.

4.1. Жизненный цикл аутентификации

Изложенные выше принципы и особенности НА лежат в основе подтверждения легитимности обучаемого при онлайн-тестировании.

Исследования возможностей НА личности легитимного пользователя выполнены на основе вычислительного эксперимента. Эксперимент проведен в соответствии со структурной схемой системы, представленной на рис. 5, и включает 3 основных подсистемы:

– регистрация;

– адаптация или обновление шаблона;

– аутентификация пользователя.

Причем при дистанционном обучении каждая подсистема работает в непрерывном режиме. Как показано на рис. 5, общими этапами и при регистрации, и при аутентификации являются: сбор данных, их обработка, извлечение клавиатурных признаков.

4.1.1. Сбор данных в эксперименте. Основой для разработки и проверки эффективности системы НА пользователей являются некоторые наборы данных о клавиатурных нажатиях. Это могут

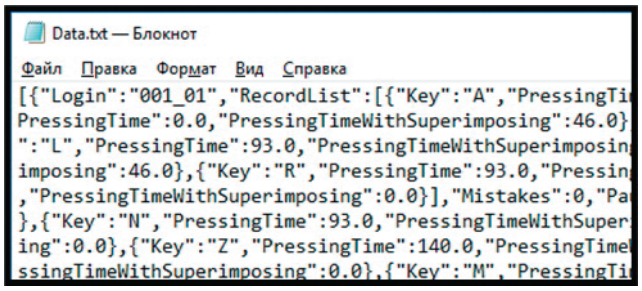

Рис. 6

быть известные статистические наборы (dataset), либо локально собранные.

Существует несколько общедоступных эталонных наборов, сформированных на основе фиксированных (Buffalo, BiosecurID) или свободных текстов (Clarkson II, Villani-2010) [23, 18, 46, 47]. Часть фиксированных наборов (CMU, WEBGREYC) [46] формируют шаблоны на основе логина/пароля, что не приемлемо для динамической аутентификации. Другая особенность подобных наборов – отсутствие шаблонов на основе русскоязычных текстов. Англоязычные шаблоны при распознавании российских пользователей представляют интерес только для первичной идентификации и статической аутентификации на основе парольных данных.

В предлагаемом исследовании осуществлен локальный сбор данных на основе разработанного программного приложения для операционной системы Windows. Архитектура программного приложения имеет клиент-серверную парадигму. На стороне клиента осуществляется локальный сбор клавиатурных данных в фоновом режиме.

Целевая аудитория данного исследования – домен университета, пользователи которого имеют навыки работы на компьютере выше среднего. Используя в ОС Windows механизм для перехвата сообщений от клавиатуры, так называемые hook-ловушки, фиксируются все события воздействия на клавиатуру. По каждому событию формируется массив информации, содержащий следующие сведения:

– логин пользователя в домене;

– код клавиши;

– событие (нажатие/отпускание клавиши);

– временная метка.

И в формате txt собранные данные имеют следующий вид (см. рис. 6).

Остальной функционал приложения реализован на сервере. Такое разделение позволяет снизить риски похищения клавиатурных данных с менее защищенных клиентских машин.

4.1.2. Предобработка данных. Собранные в течение сеанса работы пользователя сырые данные проходят в системе первичную обработку от выбросов, выходов из разумных диапазонов удержания клавиши (30–200 мс), непарных событий и коротких сессий.

Важнейшим элементом первичной обработки является выбор размера сессии (сеанса). Математически это соответствует выбору размера временных окон, на которые разбивается поток событий о клавиатурной динамике. Размер окна может быть различен на этапе регистрации и аутентификации. Методики разбиения на временные окна могут различаться [18]:

– по длительности работы пользователя или по количеству нажатий;

– окно раздвижное или скользящее.

В этом исследовании использовано скользящее окно, размер которого в режиме НА составляет 500 символов. Размер выборки при сборе данных должен быть достаточно большим, чтобы обеспечить репрезентативность отдельных букв в окне, а также статистическую значимость и несмещенность их оценок. Однако, известно, что клавиатурный почерк имеет тенденцию к изменчивости, вследствие психоэмоциональной составляющей. Поэтому временное окно должно быть скользящим. Против раздвижного окна выступает также возможность и процесса обратной биометрии: компрометация длительно незащищенных шаблонов. И объединяя эти два фактора, в работе использован минимально возможный размер выборки для состоятельной оценки средних значений для букв русского алфавита.

4.1.3. Извлечение временных характеристик и формирование клавиатурных шаблонов. Размерность результирующего пространства временных характеристик в задаче НА достаточно высока. Это связано с размером алфавита русского языка и большим числом временных характеристик КП.

Анализ исследований клавиатурных показателей продемонстрировал, что наиболее популярными временными характеристиками являются DU и UD (время удержания клавиши и пауза) и по данным [12, 20] частота использования каждого из них составляет от 30% до 40% в прикладных исследованиях.

В нашем исследовании данные о клавиатурной динамике последнего сеанса конкретного пользователя поступают на сервер в формате txt.

Далее производятся вычисления статистических характеристик DU и UD по каждой букве алфавита конкретной сессии пользователя. Результатом этой части программы является обновленный клавиатурный шаблон пользователя, адаптированный после последней сессии и сохраненный в формате *. json файла.

4.1.4. Формирование векторного показателя. Клавиатурный шаблон хранит обработанную и

достоверную информацию о клавиатурной динамике пользователей в последнем сеансе в соответствии с выбранными временными характеристиками (средние значения удержания каждой клавиши).

Для повышения информативности и достоверности созданных шаблонов при формировании совокупного показателя о клавиатурной динамике можно использовать дополнительные меры и инструменты.

Сокращение пространства на основе выделение стабильных признаков

Сокращение размерности признакового пространства временных характеристик в задаче клавиатурной динамики задача актуальная. Особенно в контексте дистанционного образования и онлайн-тестирования, когда высоки требования к скорости подтверждения легитимности экзаменуемого и обнаружения его подмены.

Выделение стабильных признаков клавиатурной динамики пользователей способствует, с одной стороны, сокращению признакового пространства. А с другой стороны – повышает селективные свойства шаблонов.

При выделении стабильных признаков можно пойти по пути повышения информативности самих временных характеристик [49]. Суть таких способов заключается в сужении диапазонов временных меток по какому-либо правилу, чаще всего с использованием статистических критериев. При этом техника сужения может касаться отдельных сеансов, отдельных пользователей и пр.

Также для повышения информативности временных характеристик могут быть использованы и эвристические способы. В данном исследовании такой способ связан с использованием частоты букв алфавита в текстах. Согласно данным Национального корпуса русского языка ruscorpora.ru частота букв (выраженная в %) уменьшается с 10.98% (буква О) до 0.996% (Х) и 0.037% (буква Ъ), рис. 7*а*. Поэтому для получения состоятельных оценок всех букв алфавита требуется внушительный размер скользящего окна клавиатурных нажатий. Что не допустимо в задаче онлайн-аутентификации. Способом сокращения размера окна является выбор в качестве стабильного признака порог частоты использования букв при наборе текстов. И для принятого в работе 0.5% порога это означает, что в шаблон пользователя не включены редко используемые буквы Ц, Щ, Э, Ф, Ъ, Ё, рис. 7*б*.

Вектор показателей КП включает характеристики (средние значения DU последнего сеанса) 27 букв русского алфавита с весовыми коэффициентами, соответствующими частотам использования букв в текстах, рис. 7*б*.

4.1.5. Распознавание легитимных пользователей. Цель статической и динамической аутентификации одинакова и состоит в подтверждении легитимности пользователя уже прошедшего первичную регистрацию. Разница только в виде создаваемого текста. При динамической (непрерывной) аутентификации – произвольный текст создается пользователем в любом приложении ОС и в произвольное время, при статической – текст и время его создания предопределены системой безопасности.

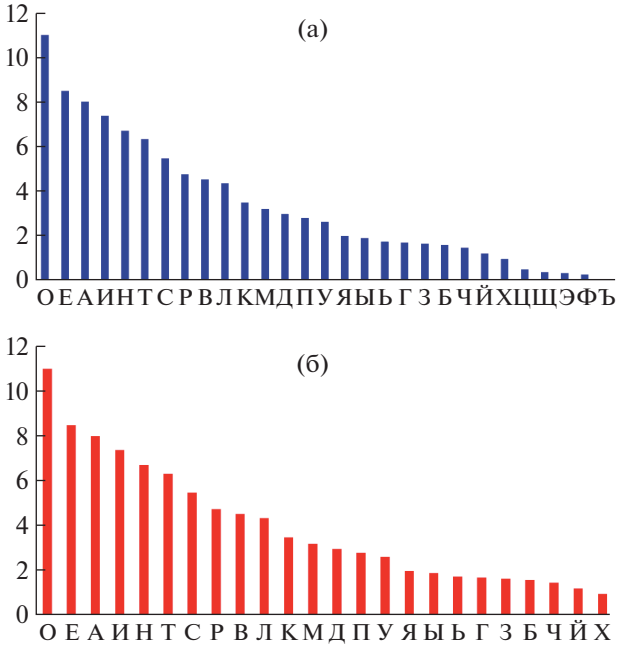

Рис. 7. Частота использования букв русского алфавита.

Следует отметить, что подтверждение легитимности пользователя по динамике его работы на клавиатуре компьютера является задачей одноклассовой классификации. Эта задача более сложна, чем многоклассовая классификация, поскольку данные нелегитимного пользователя системе распознавания неизвестны. Обучение классификатора происходит на объектах одного класса, а при тестировании алгоритм определяет принадлежность нового объекта этому классу. Данные зарегистрированных пользователей хранятся в системе в виде динамически обновляемых шаблонов. Их можно считать объектами одного класса – легитимных пользователей.

В системах онлайн-обучения одноклассовая классификация поможет выявить незарегистрированного пользователя, т.е. злоумышленника.

При онлайн-тестировании – одноклассовая классификация предотвратит несанкционированную подмену личности студента.

И хотя динамическая аутентификация значительно сложнее статической, методы распознавания аналогичны. Результаты краткого обзора ме-

Таблица 3. Сравнительный анализ показателей эффективности

	расстояние				метод	
	Евклидово		Манхэттенское		kNN	SVM
	частотность		частотность			
	–	+	–	+		
ERR, (0–100)%	10.8	0.99	10.1	0.79	0.54	3.72
порог (FAR=FRR), мс	4.8	2.3	4.8	2.2	15	
Accuracy, (0–100)%	90.87	99.20	83.41	99.4	99.45	
Precision, (0–1)	0.83	0.98	0.73	0.98	0.98	
Recall, (0–1)	0.71	0.83	0.74	0.94	0.98	

тодов динамической аутентификации на основании исследований последнего десятилетия представлены в табл. 3. Разделение методов распознавания на классы достаточно условно, но можно выделить методы машинного обучения, статистические методы и основанные на оценке метрических расстояний [20].

При выполнении этой работы в качестве классификатора были выбраны наиболее популярные методы распознавания в каждой из трех групп в рамках задачи одноклассовой классификации:

– метод опорных векторов SVM [17, 34, 21];

– метод k-ближайших соседей [27, 17, 46, 21, 34];

– оценка расстояний с использованием Евклидовой и Манхэттенской метрик.

Основная идея одноклассового метода опорных векторов OCSVM (one-class support vector machine) – выделить границы одного класса, а не разделить объекты нескольких классов, как в многоклассовой задаче [48]. Также одноклассовая классификация известна, как задача обнаружения аномалий и используется для поиска в данных не ожидаемого поведения пользователя [49]. Например, при изменении его психоэмоционального состояния под алкогольным или другим воздействием.

OCSVM отображает векторы признаков в пространство более высокой размерности с помощью функции ядра. И в случае радиального ядра (rbf kernel) находится гиперплоскость, отделяющую от начала координат большинство объектов заданного класса, в данном случае легитимного пользователя. Исключениями являются объекты, которые лежат ближе к началу координат, чем полученная гиперплоскость.

Одноклассовый метод k-ближайших соседей (One-Class KNN k-nearest neighbours) оценивает расстояние между объектами класса. Новый объект считается исключением, если большая часть объектов (например, p-я часть) всего класса находится на расстоянии, большем D, от нового объекта. Расстояние вычисляется согласно выбранной метрике в пространстве признаков, p и D – параметры метода.

Алгоритмы OCSVM и OCkNN достаточно просты в реализации. Их главное требование – это репрезентативность исходного набора данных.

5. РЕЗУЛЬТАТЫ

В соответствии с рис. 2, основными процессами жизненного цикла НА при онлайн-обучении являются этапы регистрации данных и аутентификации личности обучаемого. При регистрации производится непрерывный сбор данных о клавиатурных нажатиях и последующее извлечение показателей клавиатурной динамики, рис. 5.

Шаблоны (профили) пользователей домена динамические, сформированы они на основе непрерывного мониторинга произвольных нажатий на клавиатуру и не связаны с конкретными приложениями ОС. Сбор данных о клавиатурной динамике пользователя домена осуществляется в режиме скользящего окна и формируется шаблон с использованием букв русского и английского алфавитов. Исследования проведены на базе домена национального российского университета. Размер окна включает 500 нажатий, при накоплении которых информация передается в серверный компонент программы для предварительной очистки и извлечения показателей КП. При новых нажатиях окно передвигается, следующий набор данных передается на сервер и так происходит непрерывная регистрация данных в текущем сеансе работы пользователя.

Серверная часть программы производит расчет средних значений ВУК в текущем сеансе для каждого символа и обновляет шаблон в банке.

По каждому пользователю в банке хранятся шаблоны 10 последних сеансов, что позволяет отслеживать изменчивость КП, связанную с усталостью, психоэмоциональным состоянием и пр.

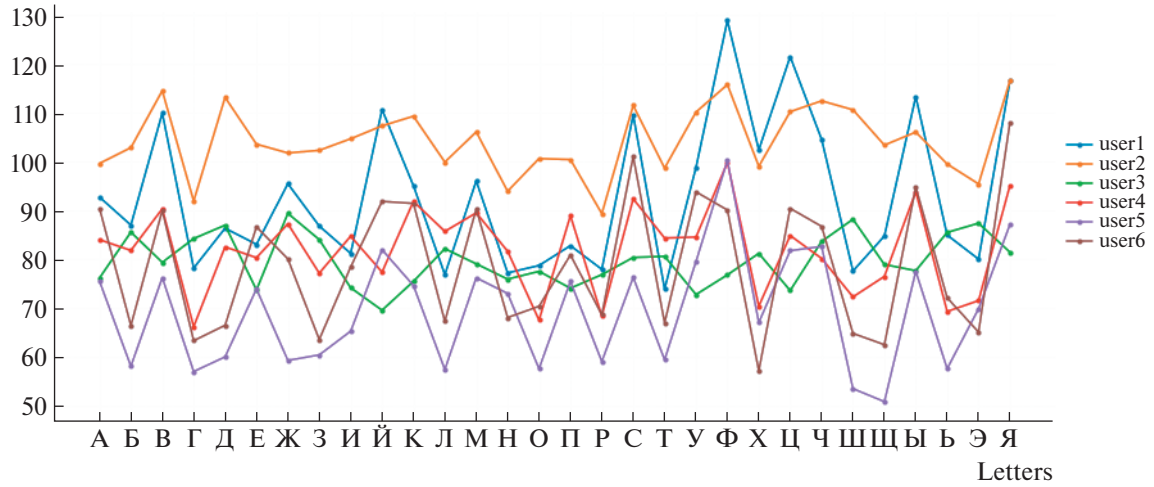

Рис. 8. Визуализация шаблонов пользователей домена.

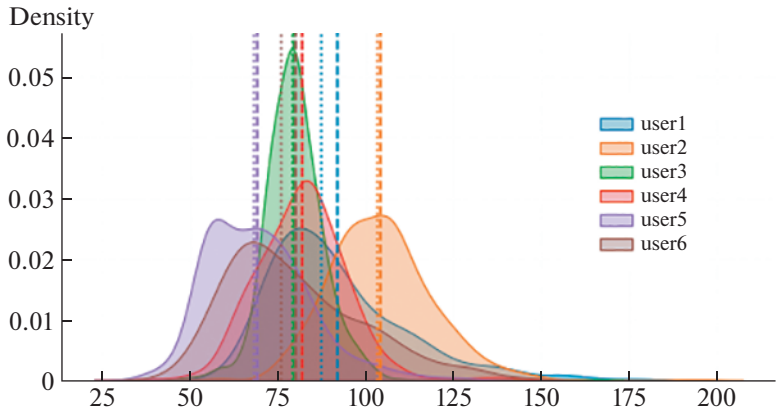

Рис. 9. Плотность распределения показателей КП.

Идентификационные возможности шаблонов разных пользователей могут быть оценены визуально и статистически. На рис. 8 изображены шаблоны шести произвольных пользователей домена университета для букв русского алфавита. Ось ординат соответствует длительности удержания клавиши в мс. Рисунок отдельной линии визуально отображает ритм набора текста и определяет, так называемый, клавиатурный почерк пользователя.

Ритм и скорость набора текстов различны для разных пользователей. Поэтому отличаются и рисунки КП. Это расхождение показателей КП подтверждается и статистически, например, видом и параметрами плотности распределения, рис. 9

Так, гистограмма пользователя User3 отличается малым разбросом и средней скоростью набора (математическое ожидание), что полностью совпадает с картиной на рис. 8 и подтверждается малыми отклонениями между модой и медианой ряда (крупный и мелкий пунктир на графиках).

Из рис. 8, 9 следует, что примерно равны и средние значения для User4 и User1. Однако распределение User1 имеет, так называемые, “тяжелые хвосты”. Мода и медиана для этого ряда значительно различаются между собой, как и для пользователя User06. И при репрезентативной выборке и высокой скорости набора это может являться дискриминантным признаком измененного психоэмоционального состояния.

Высокую скорость набора имеет также User5. При этом плотность распределения имеет бимодальный характер, что соответствует его хорошим навыкам и особенностям работы с клавиатурой. Для этого пользователя имеют место наложения при наборе текста, когда следующая клавиша нажимается, а предыдущая не отжата.

Следующий этап распознавания – подтверждение легитимности онлайн-обучаемого. Для

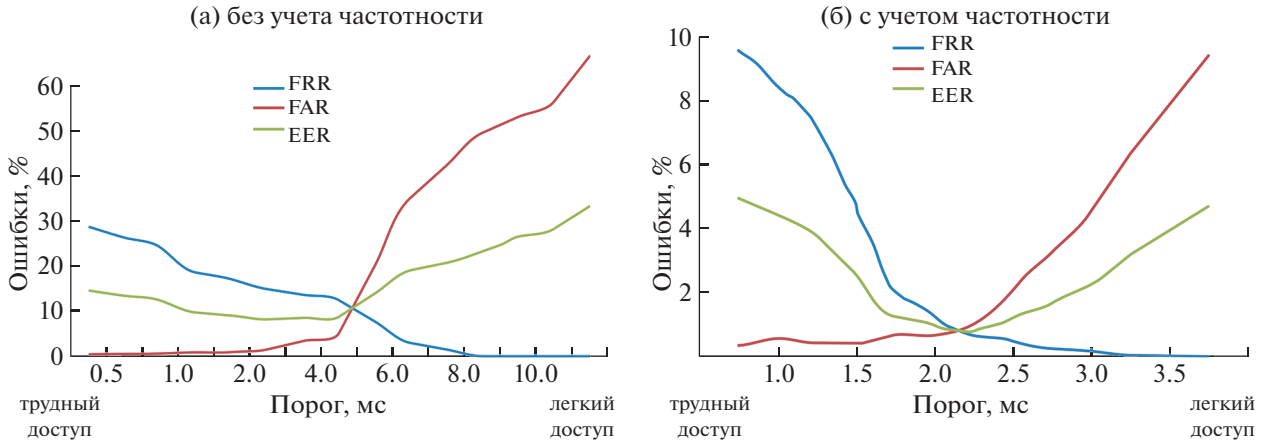

Рис. 10. Оценка эффективности распознавания пользователей.

этого производится сопоставление шаблона зарегистрированного пользователя из банка с его текущим шаблоном, рис. 5. При подтверждении легитимности личности возможны два исхода:

– шаблон последнего сеанса совпадает с шаблоном из банка данного зарегистрированного пользователя;

– шаблоны различны.

Совпадение шаблонов проанализировано в работе с использованием методов опорных векторов SVM, k-ближайших соседей и оценки расстояний на основе Евклидовой и Манхэттенской метрик. Обоснование выбора методов приведено в разделе 4.1.5.

Важнейшей характеристикой в любом методе является порог принятия решения. Пороговое значение выбирается (назначается) системой безопасности исходя из приоритета задач. Небольшие пороговые значения соответствуют малой разнице между базовым и текущем шаблоном и обеспечат сложный доступ в корпоративную сеть для всех, включая легитимных пользователей. Большой порог (низкая чувствительность) – легкий вход для всех. И основные визуальные инструменты, ошибки 1–2 рода, для Манхэттенской метрики приведены на рис. 10 в зависимости от пороговых значений.

На рис. 10*б* FAR, FRR и EER представлены с учетом частотности использования букв, 10*а* – без учета.

Как видно, достигнуто заметное уменьшение ошибок, в среднем на порядок при разных пороговых значениях. Например, для данных в табл. 3 эти значения равны 10.8% и 0.99% для Евклидовой метрики. В табл. 3 для всех выбранных методов ошибки EER приведены для порогового значения, при котором FAR = FRR.

Показатели аутентификации на основе Манхэттенской и Евклидовой метрик практически не отличаются с учетом частотности букв в алфавите. Хотя несколько отличались без использования частотности. Метод kNN показал немного лучшие показатели ERR = 0.54% при несколько большем пороговом значении 15 мс. Но это потребовало длительного подбора параметров алгоритма и не гарантирует необходимости подстройки параметров в реальных условиях. Отсутствием оптимальных значений параметров алгоритма и объясняются невысокие показатели аутентификации SVM метода.

Полезным инструментом для понимания событий при распознавании законных и незаконных пользователей является, так называемая, матрица соответствия (confusion matrix), табл. 4. На пересечении строк и столбцов матрицы показаны возможные верные (T) или ложные (F) исходы распознавания: принять (A) или отклонить (R) пользователя. Столбцы соответствуют исходу распознавания, строки – реальным пользователям.

Указанные в табл. 3 показатели точности, вычисляются следующим образом:

$$Accuracy = \frac{TA + TR}{TA + FA + TR + FR}. \quad (3)$$

$$Precision = \frac{TA}{TA + FA}. \quad (4)$$

$$Recall = \frac{TA}{TA + FR}. \quad (5)$$

Все три показателя отражают точность аутентификации легитимного пользователя. Но акценты каждого показателя разные.

Accuracy – метрика точности верных допусков и отклонений пользователя для всех возможных законных и незаконных пользователей.

Таблица 4. Матрица соответствия ошибок

		Пользователь при распознавании	
		законный	незаконный
Фактический пользователь	законный	TA	FR
	незаконный	FA	TR

Precision показывает отношение верно принятых пользователей ко всем допущенным системой.

Recall – доля допущенных пользователей из всех легитимных. Recall также называют чувствительностью модели для распознавания законных пользователей.

Accuracy принято выражать в процентах, а Precision и Recall изменяются в диапазоне (0–1).

И последний визуальный инструмент качества аутентификации – кривые DET и ROC, представленные на рис. 11.

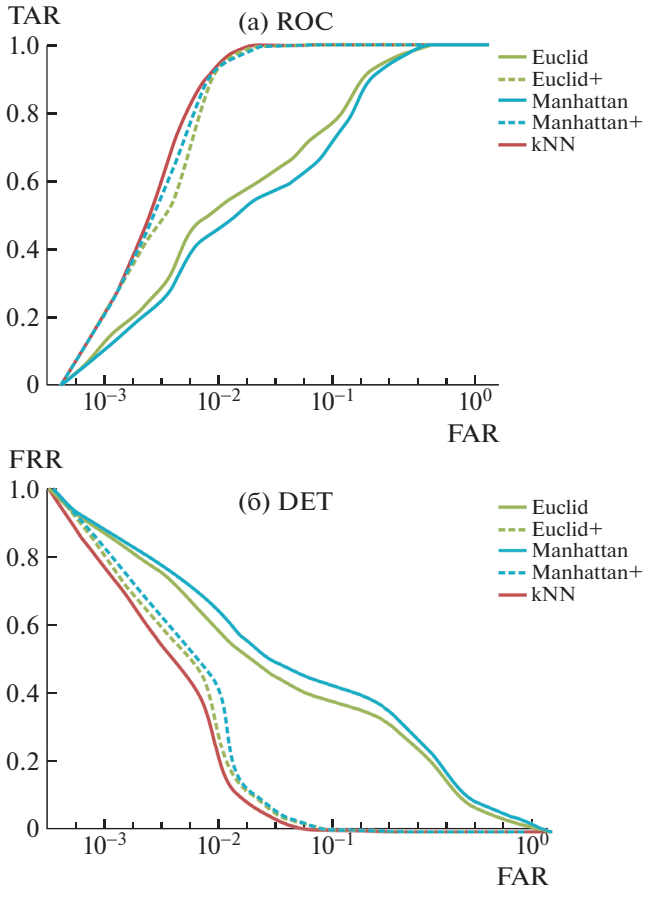

Рис. 11. Оценка эффективности распознавания пользователей.

Кривые ROC и DET подтверждают близкую эффективность методов, основанных на близости шаблонов с учетом частотности и kNN-метода.

ЗАКЛЮЧЕНИЕ

В работе рассмотрен подход к подтверждению легитимности личности при дистанционном обучении на основе мониторинга характеристик его клавиатурного почерка.

Было установлено, что скрытая идентификация обучаемого возможна на основе непрерывного мониторинга его клавиатурных нажатий в любом программном приложении. На основании проведенных исследований были сделаны следующие выводы.

1. Требуется корректировка (адаптация) образцов КП с использованием скользящего окна, позволяющая динамически отслеживать изменчивость КП пользователя и его психоэмоциональное состояние.

2. Выделение стабильных признаков клавиатурной динамики пользователей способствует и сокращению признакового пространства КП, что повышает селективные свойства шаблонов.

3. В данном исследовании таким признаком является частотность использования букв алфавита в текстах в соответствии с данными Национального корпуса русского языка ruscorpora.ru. Исключение из шаблонов шести букв с частотой ниже 0.5% привело к заметному снижению всех показателей эффективности распознавания. Например, ошибка ERR снизилась на порядок, в среднем с 10% до 1%. А показатели точности (Accuracy, Precision, Recall) в среднем повысились на 6–13% и составляют 98% как для Евклидовой, так и Манхэттенской метрики.

4. kNN-метод при оценке легитимности показал для оптимальных настроечных параметров немного лучшие результаты по ошибке ERR = = 0.54% (против 0.79%) при равных оценках точности Accuracy = 99%, Precision = 0.98.

5. Прикладной эффект дополнения простых алгоритмов оценки расстояний частотностью букв алфавита состоит в отсутствии сложных процедур настройки оптимальных параметров, как в

методах машинного обучения (kNN), при равных показателях точности.

А поскольку ключевым фактором непрерывной аутентификации легитимного пользователя является производительность распознавания, то адаптация параметров значительно снижает общую эффективность подтверждения легитимности.

СПИСОК ЛИТЕРАТУРЫ

1. *Ключевская Н.* Информационная безопасность и COVID-19. Рекомендации для бизнеса и граждан. https.//www.garant.ru/article/1421147.
2. Аналитика отрасли информационной безопасности. https.//www. infowatch.ru/analytics/analitik.
3. Хакеры атакуют университеты и колледжи. Дайджест утечек. https.//www.infowatch.ru/analytics/daydzhesty-i-obzory/khakery-atakuyut-universitety-i-kolledzhi-daydzhest-utechek.
4. Образование. От закрытия учебных заведений до возобновления их работы. https.//ru.unesco.org/covid19/educationresponse.
5. Образование: от разрушения к выздоровлению. https.//ru.unesco.org/covid19/educationresponse.
6. *Fenu G., Marras M., Boratto L.* A multi-biometric system for continuous student authentication in e-learning platforms. Pattern Recognition Letters. 2018. V. 113. P. 83–92. doi.org/. https://doi.org/10.1016/j.patrec.2017.03.027
7. *Ngqondi T., Maoneke PB., Mauwa L.* A secure online exams conceptual framework for South African universities. Social Sciences & Humanities Open. 2021. V. 3(1). 100132. DOI.org/10.1016/j.ssaho.2021.100132.
8. *Christine Lee.* How to Maintain Academic Integrity in Distance Learning. https://www.turnitin.com/ru/blog/kak-podderzhivat-akademicheskuyu-chestnost-pri-distantsionnom-obuchenii.
9. *Al-Naji F.H., Zagrouba R.* A survey on continuous authentication methods in Internet of Things environment. Computer Communications. 2020. V. 163. P. 109–133. DOI.org/10.1016/j.comcom.2020.09.006.
10. *Dasgupta D., Roy A., Nag A.* Advances in User Authentication. Springer International Publishing. 2017. DOI.org/10.1007/978-3-319-58808-7.
11. *Stylios I., Kokolakis S., Thanou O., Chatzis S.* Behavioral devices. A survey Information Fusion. 2021. V. 66. P. 76–99. DOI.org/10.1016/j.inffus.2020.08.021.
12. *Toosi R., Akhaee M.A.* Time-frequency analysis of keystroke dynamics for user authentication Future Generation. Computer Systems. 2021. V. 115. P. 438–447. DOI.org/10.1016/j.future.2020.09.027.
13. Future of identity study – IBM security Source. https.//www.ibm.com/downloads/cas/QRBY08NO.
14. *Hazan I., Margalit O., Rokach L.* Supporting unknown number of users in keystroke dynamics models. Knowledge-Based Systems. 2021. V. 221. 106982. DOI.org/10.1016/j.knosys.2021.106982.
15. *Parkinson S., Khan SCrampton A., Xu Q., Xie W., Liu N., Dakin K.* Password policy characteristics and keystroke biometric authentication. IET Biometrics. 2021. V. 10(2). P. 163–178. DOI.org/10.1049/bme2.12017.
16. *Kochegurova E.A., Gorokhova E.S., Mozgaleva A.I.* Development of the Keystroke Dynamics Recognition System. J. Physics. Conf. Ser. 2017. V. 803. 012073. DOI.org/10.1088/1742-6596/803/1/012073.
17. *Kim J., Kim H., Kang P.* Keystroke dynamics-based user authentication using freely typed text based on user-adaptive feature extraction and novelty detection. Applied Soft Computing. 2018. V. 62. P. 1077–1087. DOI.org/10.1016/j.asoc.2017.09.045.
18. *Lu X., Zhang S., Hui P., Lio P. Continuous* authentication by free-text keystroke based on CNN and RNN. Computers & Security 2020. V. 96. 01861. DOI.org/10.1016/j.cose.2020.101861.
19. *Dargan S., Kumar M.* A comprehensive survey on the biometric recognition systems based on physiological and behavioral modalities. Expert Systems with Applications. 2020. V. 143. 113114. DOI. org/. https://doi.org/10.1016/j.eswa.2019.113114
20. *Kochegurova E.A., Martynova Y.A.* Aspects of continuous user identification based on free texts and hidden monitoring. Program Comput Softw. 2020. V. 46(1). P. 12–24. DOI. https://doi.org/10.1134/S036176882001003X
21. *Zaidi A.Z., Chong C.Y. Jin Z., Parthiban R., Sadiq A.S.* Touch-based continuous mobile device authentication. State-of-the-art, challenges and opportunities. J Network Comput Appl. 2021. V. 191. 103162. DOI.org/10.1016/j.jnca.2021.103162.
22. *Teh P.S., Teoh A.B.J., Yue S.* A survey of keystroke dynamics biometrics The Scientific World Journal. 2013. V. 2013. P. 1–24. DOI. https://doi.org/10.1155/2013/408280
23. *Morales A., Fierrez J., Tolosana R., Ortega-Garcia J., Galbally J., Gomez-Barrero M., Anjos A., Marcel S.* KBOC. Keystroke biometrics OnGoing competition. 2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS). 2016. DOI. https://doi.org/10.1109/BTAS.2016.7791180.
24. *Pisani P.H., Lorena A.C. A* systematic review on keystroke dynamics. J Braz Comput Soc. 2013. V. 19(4). P. 573–587.
25. *Gunetti D., Picardi C.* Keystroke analysis of free text. ACM Trans Inf Syst Secur. 2005. V. 8(3). P. 312–347. DOI. https://doi.org/10.1145/1085126.1085129
26. *Kochegurova E., Luneva E., Gorokhova E.* On continuous user authentication via hidden free-text based monitoring. Adv Intell Sys Comput. 2019. V. 875. P. 66–75. DOI. https://doi.org/10.1007/978-3-030-01821-4_8
27. *Alsultan A., Warwick K.* Keystroke dynamics authentication. a survey of free-text methods. Int. J. Comput. Sci. 2013. V. 10(4). P. 1–10.
28. *Mondal S., Bours P.* A study on continuous authentication using a combination of keystroke and mouse biometrics. Neurocomputing. 2017. V. 230. P. 1–22. DOI. https://doi.org/10.1016/j.neucom.2016.11.031
29. *Zhong Y., Deng Y.* A survey on keystroke dynamics biometrics. approaches, advances, and evaluations. Recent Advances in User Authentication Using Keystroke Dy-

namics Biometrics. 2015. V. 2. P. 1–22. DOI. https://doi.org/10.15579/gcsr.vol2.ch1

30. *Ali M.L., Monaco J.V., Tappert C.C. et al.* Keystroke Biometric Systems for User Authentication. J Sign Process Syst. 2017. V. 86. P. 175–190. https://doi.org/10.1007/s11265-016-1114-9

31. *Alsultan A., Warwick K., Wei H.* Non-conventional keystroke dynamics for user authentication. Pattern Recogn Lett 2017. V. 89. № 5. P. 53–59. DOI. https://doi.org/10.1016/j.patrec.2017.02.010

32. *Shimshon T., Moskovitch R., Rokach L., Elovici Y.* Continuous verification using keystroke dynamics. International Conference on Computational Intelligence and Security (CIS'10). 2010. P. 411–415. DOI. https://doi.org/10.1109/CIS.2010.95.

33. *Messerman T., Mustafić T., Camtepe S.A., Albayrak S.* Continuous and non-intrusive identity verification in real-time environments based on free-text keystroke dynamics. 2011 International Joint Conference on Biometrics (JCB). 2011. P. 1–8. DOI. https://doi.org/10.1109/IJCB.2011.6117552.

34. *Chang H.C., Li J., Wu C., Stamp M.* Machine Learning and Deep Learning for Fixed-Text Keystroke Dynamics. arXiv.2107.07409v1 [cs.LG]. 2021. DOI.org/10.48550/arXiv.2107.07409.

35. *Ahmed A.A., Traore I.* Biometric recognition based on free-text keystroke dynamics/ Cybern. IEEE Trans 2014. V. 44(4). P. 458–472. DOI. https://doi.org/10.1109/TCYB.2013.2257745

36. *Goodkind A., Brizan D.G., Rosenberg A.* Utilizing overt and latent linguistic structure to improve keystroke-based authentication. Image and Vision Computing 2017. V. 58. P. 230–238. DOI. https://doi.org/10.1016/j.imavis.2016.06.003

37. *Al Solami E., Boyd C., Clark A., Ahmed I.* User-representative feature selection for keystroke dynamics. 5th International Conference on Network and System Security (NSS'11) 2011. P. 229–233. DOI. https://doi.org/10.1109/ICNSS.2011.6060005.

38. *Eberz S., Rasmussen K.B., Lenders V., Martinovic I.* Evaluating behavioral biometrics for continuous authentication. challenges and metrics. 2017 ACM on Asia Conference on Computer and Communications Security (ASIA CCS '17). 2017. P. 386–399. DOI. https://doi.org/10.1145/3052973.3053032.

39. *Antal M., Szabó L.Z., Laszlo I.* Keystroke dynamics on Android platform. Procedia Technology. 2015. V. 19. P. 820–826. DOI. https://doi.org/10.1016/j.protcy.2015.02.118

40. *Locklear H., Govindarajan S., Sitova Z. etc.* Continuous authentication with cognition-centric text production and revision features. IEEE/IAPR international joint conference on biometrics (IJCB 2014). 2014. DOI. https://doi.org/10.1109/BTAS.2014.6996227

41. *Kang P., Cho S.* Keystroke dynamics-based user authentication using long and free text strings from various input devices. Inf Sci. 2015. V. 308. P. 72–93. DOI. https://doi.org/10.1016/j.ins.2014.08.070

42. *Matsubara Y., Samura T., Nishimura H.* Keyboard Dependency of Personal Identification Performance by Keystroke Dynamics in Free Text Typing. Journal of Information Security. 2015. V. 6. P. 229–240. DOI. https://doi.org/10.4236/jis.2015.63023

43. *Wang X., Yan Z., Zhang R., Zhang P.* Attacks and defenses in user authentication systems. A survey. Journal of Network and Computer Applications. 2021. V. 188. 103080. DOI. https://doi.org/10.1016/j.jnca.2021.103080

44. *Muzaffar A.W., Tahir M., Anwar M.W., Chaudry Q., Mir S.R., Rasheed Y.* A systematic review of online exams solutions in e-learning. Techniques, tools, and global adoption. IEEE Access. 2021. V. 9. P. 32689–32712. DOI. https://doi.org/10.1109/ACCESS.2021.3060192

45. *Jagadamba G., Sheeba R., Brinda K.N., Rohini K.C., Pratik S.K.* Adaptive E-Learning Authentication and Monitoring. 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA). 2020. 277–283. DOI. https://doi.org/10.1109/ACIMIA48430.2020.9074955.

46. *Iapa A., Cretu V.* Shared Data Set for Free-Text Keystroke Dynamics Authentication Algorithms. Preprints. 2021. 2021050255. DOI. https://doi.org/10.20944/preprints202105.0255.v1

47. *González N., Calot E.P., Ierache J.S., Hasperué W.* On the shape of timings distributions in free-text keystroke dynamics profiles. Heliyon 2021. V. 7(11). e08413. DOI. https://doi.org/10.1016/j.heliyon.2021.e08413

48. *Mhenni A., Cherrier E., Rosenberger C., Essoukri Ben Amara N.* Analysis of Doddington zoo classification for user dependent template update. Application to keystroke dynamics recognition. Future Gener Comput Syst. 2019. V. 97. P. 210–218. DOI. . https://doi.org/10.1016/j.future.2019.02.039

49. *Казачук М.А.* Динамическая аутентификация пользователей на основе анализа работы с клавиатурой компьютера. Дисс. на соискание уч. степени к.ф.-м.н. Москва. 2019. 155 с.

50. *Alpar O.* Biometric keystroke barcoding. A next-gen authentication framework. Expert Sys Appl. 2021. V. 177. 114980. DOI. https://doi.org/10.1016/j.eswa.2021.114980

51. *Yang Y., Guo B., Wang Z., Li M., Yu Z., Zhou X.* BehaveSense. Continuous authentication for security-sensitive mobile apps using behavioral biometrics. Ad Hoc Networks. 2019. V. 84. P. 9–18. DOI. https://doi.org/10.1016/ j.adhoc.2018.09.015

EDN: OZREFS
ПРОГРАММИРОВАНИЕ, 2022, № 6, с. 46–53

ПАРАЛЛЕЛЬНОЕ И РАСПРЕДЕЛЕННОЕ ПРОГРАММИРОВАНИЕ

УДК 681.3

ПОСТРОЕНИЕ БОРТОВОЙ КОММУТИРУЕМОЙ СЕТИ С ВРЕМЕННОЙ СИНХРОНИЗАЦИЕЙ МИНИМАЛЬНОЙ СЛОЖНОСТИ

© 2022 г. В. А. Костенко[a,*] (ORCID: 0000-0002-7895-2322),
А. А. Морквин[a,] (ORCID: 0000-0001-6253-054X)**

[a] Московский государственный университет имени М.В. Ломоносова
Кафедра автоматизации систем вычислительных комплексов
119992, Россия, Москва, Ленинские горы, д. 1, строение 52

**E-mail: kost@cs.msu.su*

***E-mail: mr.andrej1102@yandex.ru*

Поступила в редакцию 12.05.2022 г.
После доработки 16.07.2022 г.
Принята к публикации 20.07.2022 г.

В статье сформулирована задача построения бортовой коммутируемой сети с временной синхронизацией минимальной сложности необходимой для передачи периодических сообщений в реальном времени и предложены алгоритмы ее решения: построения структуры сети и системы расписаний передач сообщений. Приводятся результаты апробации предложенных алгоритмов для построения бортовых сетей на основе стандарта Time-Sensitive Networking (TSN).

DOI: 10.31857/S013234742206005X

1. ВВЕДЕНИЕ

Современные информационно управляющие системы реального времени (ИУС РВ) являются распределенными и включают в свой состав различные устройства, которые взаимодействуют между собой. В ИУС РВ можно выделить три уровня обработки данных: 0) уровень предобработки, 1) уровень первичной обработки, 2) уровень вторичной обработки. Для обеспечения выполнения функциональных программ в режиме реального времени наибольшая производительность и пропускная способность сети необходима на уровне первичной обработки данных.

В настоящее время осуществляется переход от ИУС РВ с федеративной архитектурой к ИУС РВ с интегрированной модульной архитектурой. Наиболее широко используемый подход к построению ИУС РВ с интегрированной модульной архитектурой известен как интегрированная модульная авионика (ИМА). Разработан ряд стандартов, регламентирующих построение ИУС РВ с архитектурой ИМА: ARINC 651 – основные принципы построения ИУС РВ на основе ИМА [1]; ARINC 653 – спецификация операционных систем [2]; FC-AE-ASM-RT – спецификация сети информационного обмена на основе коммутируемой сети Fibre Channel [3]; ARINC 664 (AFDX) – спецификация сети информационного обмена на основе Ethernet [4].

Множество стандартов Time-Sensitive Networking (TSN) [5] является расширением стандарта IEEE 802.1 Audio Video Bridging (AVB) [6], который был впервые опубликован в 2005 году, и разработано для управления сетевым трафиком, чтобы обеспечивать строго определенные временные задержки при передаче данных. Для этого все устройства в TSN сети должны быть синхронизированы друг с другом относительно единого эталона времени для поддержания связи в реальном времени. На данный момент множество стандартов и его спецификации под конкретные области использования продолжают активно разрабатываться. Однако уже есть ИУС РВ, сети обмена которых построены на основе этих стандартов: ИУС РВ автоматизированной производственной линии [7] и сеть беспилотного автомобиля [8]. Гарантия низких сквозных задержек делают TSN сети актуальными [9] для использования в ИУС РВ.

В работе [10] на примере локационной системы с фазированными антенными решетками показано, что переход от федеративных архитектур к архитектурам ИМА (то есть при переносе программ первичной обработки с вычислительной системы на единый бортовой вычислитель) приводит к увеличению потока данных в бортовой сети обмена в 10^3–10^5 раз в зависимости от характеристик локационной системы. Минимизация сложности сети с сохранением способности пере-

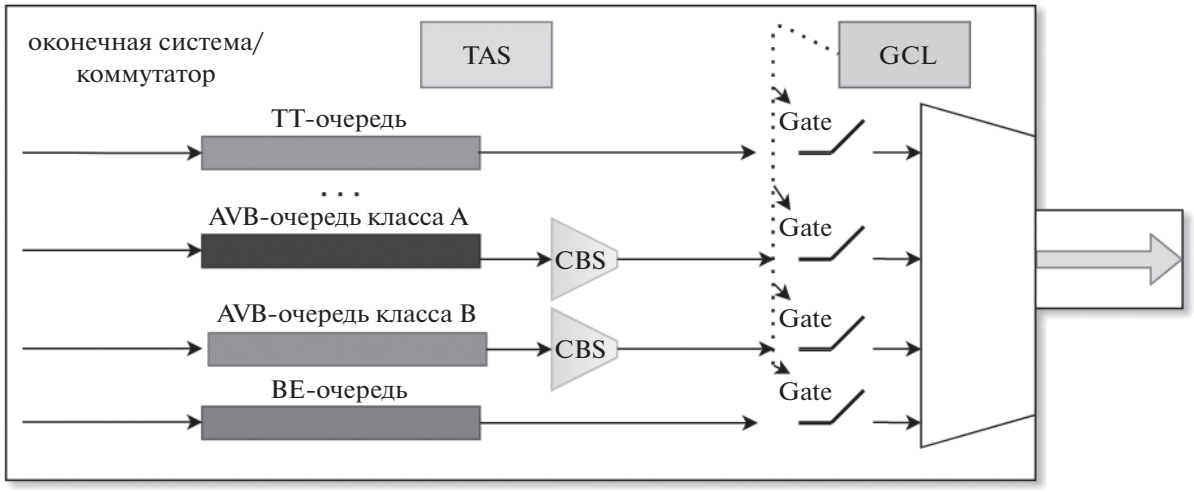

Рис. 1. Time-Aware Shaper (*TAS*).

давать заранее заданный трафик реального времени является актуальной задачей проектирования ИУС РВ с архитектурой ИМА. С точки зрения надежности, чем меньше суммарная длина кабелей в сети, тем меньше перекрестные наводки, и, следовательно, меньше вероятность ошибок при передаче данных [11].

В данной работе сформулирована задача построения бортовой сети обмена минимальной сложности (минимизируется суммарная длина физических соединений) на основе стандарта TSN необходимой для передачи в реальном времени заранее заданного множества периодических сообщений и предложены алгоритмы ее решения: построения структуры сети и системы расписаний передач. Приводятся результаты апробации предложенных алгоритмов для построения бортовых сетей.

2. ОПИСАНИЕ TSN-СЕТЕЙ

TSN-сеть определяет механизм передачи трафика с требованиями к длительности передачи на основе Ethernet сетей. Ключевой особенностью является временная синхронизация всех устройств в сети на основе протокола Precision Time Protocol (PTP), который позволяет синхронизировать внутренние часы каждого устройства с разницей от 10 нс до 1 мкс. Такая точность достигается за счет “аппаратной” поддержки синхронизации, которая точно учитывает задержки. Также TSN-сеть позволяет осуществлять надежную доставку сообщений за счет возможности использования нескольких маршрутов передачи сообщений в сети.

Согласно [5] существует три вида трафика в сети:

1. Time-Triggered traffic (*TT*-трафик) – трафик с гарантией низких сквозных задержек и доставки без потерь.

2. Audio-Video-Bridging traffic (*AVB*-трафик) – трафик с ограниченными сквозными задержками. Данный тип трафика является менее критичным, чем *TT*-трафик, и состоит из двух подтипов – класса *A* и класса *B*, которые отличаются приоритетом.

3. Best-Effort traffic (*BE*-трафик) – негарантированный трафик без гарантий доставки и ограничений на задержку.

Рассмотрим подробнее вопрос планирования передачи сообщения. На рисунке 1 изображен планировщик Time-Aware Shaper (*TAS*), который ассоциируется с каждым выходным портом коммутатора. Согласно стандарту [5] для планирования передачи доступно восемь очередей (с номерами соответственно от 1 до 8), одна и более могут быть зарезервированы под *TT*-трафик, две соответственно под класс *A* и *B AVB*-трафика, оставшиеся могут быть зарезервированы под *BE*-трафик.

С каждой очередью ассоциированы “задвижки” (*Gate*), которые могут находиться в двух состояниях – открыты или закрыты. Состояние в каждый момент времени определяется благодаря наличию расписания Gate Control List (*GCL*), которое представляет собой циклическое расписание открытия и закрытия всех *Gate* на выходном порту и длительность соответствующего открытия или закрытия. Пример *GCL* можно увидеть на рисунке 2, где T_i – длительность интервала, а битовый массив представляет собой состояние соответствующих *Gate* в данный промежуток времени.

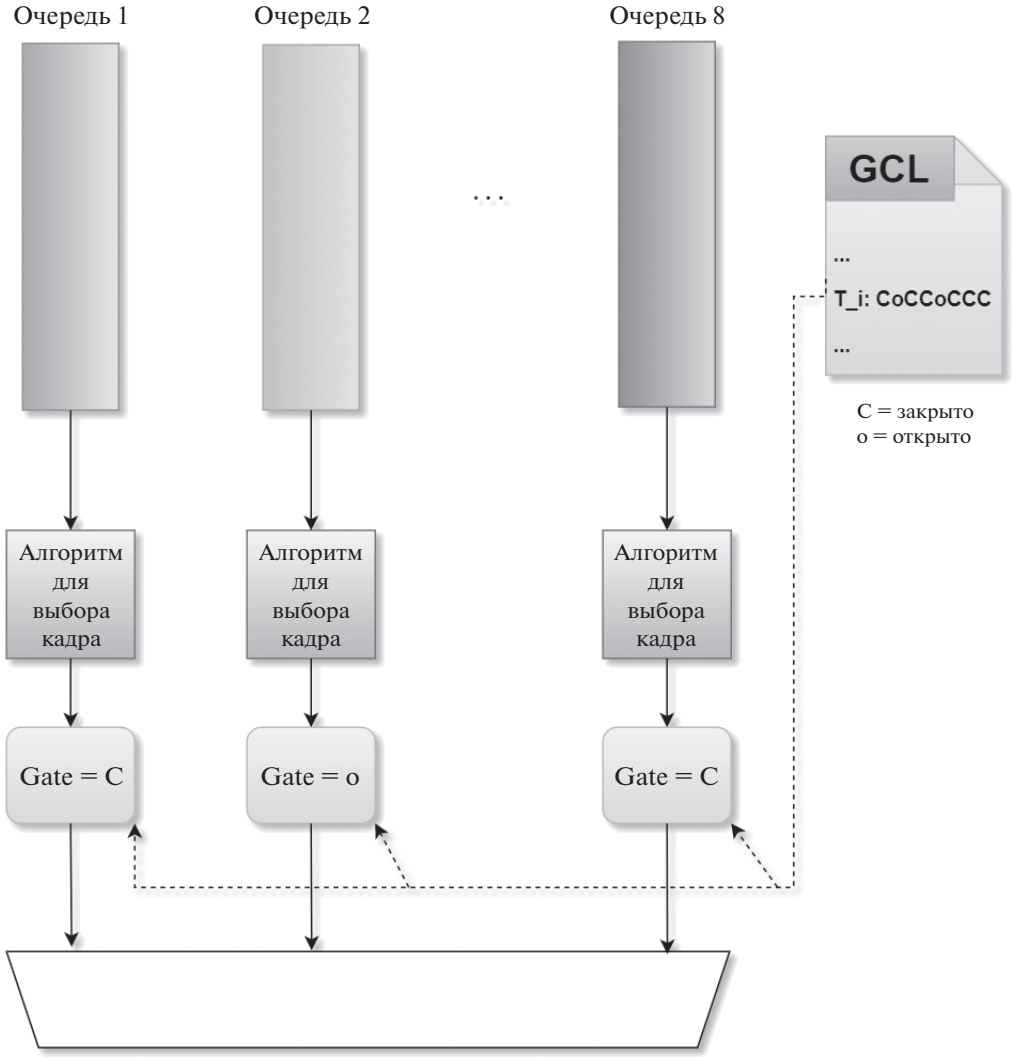

Рис. 2. Пример *GCL*.

Планировщик *TAS* при выборе кадра для передачи руководствуется следующим правилом – выбирается кадр из очереди с наивысшим приоритетом (соответственно с наименьшим номером) среди всех очередей, у которых *Gate* открыты и которые не пусты. Однако для проверки доступности кадров *AVB*-трафика *TAS* использует механизм, про который будет написано далее.

При открытии очереди *TT*-трафика может возникнуть ситуация наложения передачи кадров. Стандартом [5] предусмотрено два механизма предотвращения этой проблемы:

1. Встраивание защитных интервалов перед каждым открытием *TT*-очереди. Размер интервала зависит от максимального размера кадра других сообщений и в худшем случае выставляется для 1542 байт. Во время этого интервала запрещено начинать передачу кадров.

2. Прерывание передачи с последующим продолжением с места прерывания. В данном случае для второй части кадра добавляется дополнительный заголовок в 24 байта, который требуется для склейки кадра.

Для передачи *AVB*-трафика используется механизм Credit-Base Shaper (*CBS*), который допускает выборку кадра из очереди при выполнении следующих условий:

1. Доступен кредит для передачи.

2. Не передаются кадры с более высоким приоритетом.

Этот механизм необходим для предотвращения остановки потоков с более низким приоритетом. Кредит уменьшается на величину $sendSlop\left(\frac{байт}{с}\right)$ при передаче кадра соответствующей очереди, а увеличивается при ожидании передачи на вели-

чину $idleSlop\left(\frac{\text{байт}}{\text{с}}\right)$. При закрытии *Gate* кредит замораживается. Передача возможна только при неотрицательном кредите.

3. МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ

Введем следующие множества: N – множество точек размещения оконечных систем, K – множество возможных точек размещения коммутаторов, E_{sw} – множество возможных соединений между коммутаторами, E_{end} – множество возможных соединений между оконечными системами и коммутаторами, V – множество длин соединений. Для каждого соединения $e \in E_{sw} \cup E_{end}$ задана пропускная способность R_e. Для каждой оконечной системы $n \in N$ задано множество абонентов A_n, которые к ней подключены (один абонент может быть подключен только к одной оконечной системе).

Нагрузка на сеть задается *множеством периодически передаваемых сообщений MSG*, в котором каждое сообщение характеризуется следующими параметрами:

1. $type_{msg} \in \{TT, A, B\}$ – принадлежность сообщения к определенному типу трафика (либо TT-сообщение, либо AVB-сообщение класса A или B).

2. T_{msg} (мс) – период передачи сообщения.

3. $size_{msg}$ (байт) – размер сообщения.

4. src_{msg} – абонент-отправитель.

5. $\{dst\}_{msg}$ – множество абонентов-получателей, подключенных к оконечным системам.

Для передачи сообщений в реальном времени должны выполняться следующие условия:

1. Сообщение должно передаваться не менее одного раза в период.

2. t_{msg} (мс) – максимальная длительность передачи сообщения с момента начала периода до момента получения всеми абонентами-получателями не превосходит данное значение.

Пусть $G = (N \cup K, E, V)$ – *максимальная сеть*, которая включает в себя все коммутаторы из множества K и все соединения из множества $E = E_{sw} \cup E_{end}$. Предлагается решать задачу минимизации сложности сети путем нахождения подмножества максимальной сети достаточного для передачи множества MSG.

Сеть $G^* = (N^* \cup K^*, E^*, V^*)$ – является *подсетью* максимальной сети G: $N^* = N$ – множество оконечных систем (нельзя исключить оконечную систему без уменьшения множества передаваемых сообщений); $K^* \subseteq K$ – подмножество коммутаторов; $E^* \subseteq E$ – подмножество соединений; $V^* \subseteq V$ – длина всех соединений $e \in E^*$.

Введем понятие *меры сложности сети S* для бортовой сети обмена, как суммарную длину всех соединений:

$$S(G^*) = \sum_{e \in E^*} V^*(e)$$

Далее под минимальной сетью будем понимать сеть с минимальным значением S.

Построить расписание передачи множества сообщений MSG^* для сети G^* означает следующее: для каждого выходного порта коммутатора $k \in K^*$ построить GCL так, что будут выполняться ограничения на длительность передачи для каждого $msg \in MSG^*$. При этом важно понимать, что каждое сообщение из MSG^* имеет свой маршрут передачи. При построении GCL будем соблюдать следующее ограничение:

1. В любой момент времени в каждой TT-очереди могут находиться кадры только одного сообщения. Это ограничение вводится для изоляции потоков кадров разных сообщений, которые могут поступить на входной порт коммутатора в одно и то же время. Из-за этого может возникать джиттер и нарушаться длительность доставки.

Решение задачи заключается в построении минимальной сети G^* и расписания передач для максимального подмножества $MSG^* \subseteq MSG$, которое может быть передано через максимальную сеть.

Таким образом, имеется два критерия для оптимизации: $\max(|MSG^*|)$ и $\min(S(G^*))$. Чтобы сделать задачу оптимизации однокритериальной, оставим один критерий $\min(S(G^*))$, а другой преобразуем в ограничение $|MSG^*| = |MSG^{max}|$, где MSG^{max} – это максимальное множество сообщений для максимальной сети.

Следовательно, задача оптимизации формулируется следующим образом:

$$\min(S(G^*))$$
$$G^* \subseteq G; \quad |MSG^*| = |MSG^{max}|;$$
$$\forall msg \in MSG^* : Dur(msg) \le t_{msg}$$

Здесь $Dur(msg)$ – значение длительности передачи сообщения, которое может быть оценено по методам в следующих работах [12, 13].

4. АЛГОРИТМ ПОСТРОЕНИЯ TSN-СЕТИ МИНИМАЛЬНОЙ СЛОЖНОСТИ

Общая схема алгоритма имеет следующие шаги:

Шаг 1. Из входных данных создать максимальную сеть G.

Шаг 2. Отсортировать множество сообщений MSG (по убыванию требуемой пропускной способности).

Шаг 3. Для каждого сообщения $msg \in MSG$:

Шаг 3.1. Выполнить процедуру построения маршрута и получить маршрут, который обладает минимальной суммарной длиной используемых соединений. В случае неуспеха перейти к следующему сообщению.

Шаг 3.2. Выполнить процедуру построения расписания передачи сообщения. В случае успеха перейти на шаг 3.4.

Шаг 3.3. Выполнить процедуру ограниченного перебора для маршрутов уже назначенных сообщений, которая заключается в попытках поиска других маршрутов передачи для уже назначенных сообщений. В случае неуспеха перейти к следующему сообщению.

Шаг 3.4. Выполнить процедуру оценки длительности доставки сообщения. В случае успеха перейти к следующему сообщению.

Шаг 3.5. Выполнить процедуру ограниченного перебора соединений в маршруте сообщения, которая заключается в попытке “обхода” соединений с большим временем ожидания передачи кадров сообщения. В случае неуспеха перейти к другому сообщению.

Шаг 4. Удалить из максимальной сети G неиспользуемые соединения (соединения, через которые не проходит ни один маршрут) и неиспользуемые коммутаторы (коммутаторы без подключенных соединений). Полученная сеть является результатом работы алгоритма.

Процедура построения маршрута

Маршрут представляет собой дерево, являющееся подграфом максимальной сети G, корнем дерева является оконечная система-отправитель, листьями – оконечные системы-получатели сообщения.

Для построения маршрута выполняются следующие шаги:

Шаг 1. Если в максимальной сети между оконечной системой-отправителем и хотя бы одной из оконечных систем-получателей не существует пути, то процедура возвращает неуспех.

Шаг 2. Запустить алгоритм Дейкстры от оконечной системы-отправителя. Остановить алгоритм после нахождения кратчайшего пути до одной из оконечных систем-получателей, до которых еще не найден путь. Будем использовать следующий критерий веса для ребра – $I(e) = \frac{V_e}{k+1}$, где $e \in E$, V_e – длина соединения e, а k – число уже назначенных на это соединений сообщений.

Шаг 3. Если найдены пути до всех получателей, то вернуть объединение найденных маршрутов. Иначе запомнить путь до найденного получателя и перейти на шаг 2.

Процедура построения расписания передачи сообщения

Для каждого узла (либо оконечная-система отправитель, либо коммутатор) в маршруте сообщения выполнить следующие шаги:

Шаг 1. Для каждого участвующего в передачи сообщения выходного порта узла маршрута добавить в GCL передачу этого сообщения. В случае невозможности добавить в GCL-передачу этого сообщения вернуть неуспех, иначе перейти к другому узлу маршрута.

Процедура ограниченного перебора маршрутов уже назначенных сообщений

Шаг 1. Для каждого множества назначенных сообщений мощности не больше заданного значения m (то есть рассматриваются всевозможные сочетания из множества назначенных сообщений по m элементов) выполнить шаги 1.1–1.5:

Шаг 1.1. Снять все маршруты и расписания передач сообщений из выбранного множества.

Шаг 1.2. Выполнить процедуру построения расписания передачи сообщения для msg (см. шаг 3 общей схемы алгоритма). В случае неуспеха перейти на шаг 1.5.

Шаг 1.3. Выполнить процедуру построения маршрута, процедуру построения расписания передачи и процедуру оценки длительности доставки сообщения для всех снятых сообщений. Если хотя бы одно из сообщений не может быть назначено, то снять сообщение msg и перейти на шаг 1.5.

Шаг 1.4. Выполнить назначения сообщений согласно найденным маршрутам и построенным расписаниям. Вернуть успех.

Шаг 1.5. Восстановить изначальные назначения маршрутов и расписаний сообщений. Перейти к другому множеству сообщений.

Шаг 2. Если после ограниченного перебора построить расписание для msg не удается, то назначение сообщения считается неуспешным.

Параметр m – входной параметр алгоритма.

Процедура ограниченного перебора соединений в маршруте сообщения

Шаг 1. Сформировать $\{Paths_i\}$, $i = \overline{1, p}$ из маршрутов, которые определяются следующим образом ($Paths_i$ – i-минимальный маршрут (под минимальность понимаем сумму критериев веса ребер в маршруте), а p – заранее заданное число):

Шаг 1.1. Для каждого множества соединений из маршрута сообщения мощности не больше заданного значения h, то есть рассматриваются все-

возможные сочетания из множества соединений маршрута по h элементов:

Шаг 1.1.1. Принять длины соединений из выбранного множества за достаточно большое число.

Шаг 1.1.2. Выполнить процедуру построения маршрута для данного сообщения msg. Сохранить маршрут в $\{Paths_i\}$, если новый маршрут является новым i-минимумом.

Шаг 1.1.3. Восстановить изначальные длины соединений из выбранного множества. Перейти к следующему множеству.

Шаг 2. Для каждого маршрута из $\{Paths_i\}$:

Шаг 2.1. Выполнить процедуру построения расписания передачи для данного сообщения msg. В случае успеха перейти на шаг 2.3.

Шаг 2.2. Выполнить процедуру ограниченного перебора маршрутов уже назначенных сообщений. В случае неуспеха перейти к другому маршруту.

Шаг 2.3. Выполнить процедуру оценки длительности доставки сообщения msg. В случае успеха вернуть новый маршрут и новое расписание передачи.

Шаг 3. Если после ограниченного перебора добиться выполнения ограничения на длительность доставки для msg не удается, то назначение сообщения считается неуспешным.

Параметры p и h – входные параметры алгоритма.

Процедура оценки длительности доставки сообщения

Длительность передачи сообщения msg вычисляется следующим образом:

$$Dur(msg) = \mu + \Delta$$

Где:

• μ – константа, описывающее время, требуемое на разбиение сообщение на кадры и сборку сообщения из кадров (и, возможно, прохождение сообщения через некоторые обязательные этапы обработки, занимающие не более чем некоторое известное постоянное время);

• Δ – максимальная длительность передачи последнего кадра.

Для TT-трафика величина Δ может быть вычислена при построении расписания передач для данного сообщения, так как для TT-трафика на каждом порту известно время получения первого кадра и время ухода последнего кадра из очереди.

Для AVB-трафика величина Δ может быть вычислена с помощью одного из методов в работах [12, 13]. Рекомендуется использовать метод из работы [12], так как в данной работе учитывается влияние TT-трафика на длительность передачи AVB-трафика, поэтому оценка получается точнее, чем в других работах.

Процедура выполняет проверку ограничения на длительность передачи сообщения и возвращает успех или неуспех.

5. ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ СВОЙСТВ АЛГОРИТМА

Цель экспериментального исследования – проверить эффективность предложенного алгоритма по критерию минимизации сложности сети. При этом по сети должны передаваться в реальном времени все изначально заданные периодические сообщения. Для достижения этой цели достаточно формировать такие наборы сообщений, для которых известно, какая подсеть достаточна для передачи всех сообщений.

Наборы данных для экспериментов строились следующим образом:

1. Выбирается базовая топология сети.

2. Создается набор сообщений, который может быть передан через базовую сеть, создавая достаточно высокую нагрузку на сеть.

3. Базовая сеть расширяется до максимальной сети с дополнительными коммутаторами и соединениями так, чтобы для передачи сообщений было доступно несколько дополнительных маршрутов.

Работа алгоритма считается успешной, если для такого набора сообщений максимальная сеть была сокращена до базовой сети или до другой сети со значением сложности равным сложности базовой сети.

Три пары (базовая; максимальная) сетевых топологий, которые использовались в исследовании, изображены на рисунке 3 и 4. Сплошные квадраты, круги и линии изображают соответственно коммутаторы, оконечные системы и соединения базовой сети. Вместе с пунктирными они образуют максимальную сеть. Третья топология (рисунок 4) взята из [14] и сокращена до базовой, удалив некоторые избыточные связи между коммутаторами. Топологии типичны для бортовых сетей, в частности, третья используется в Airbus A380.

Для каждой из трех топологий сети были сгенерированы наборы сообщений следующих типов:

1. Размер: 16 Байт – 1 КБ; период: 10 мс – 1 с; максимальная длительность доставки: 10 мс – 100 мс.

2. Размер: 1 КБ – 100 КБ; период: 100 мс – 10 с; максимальная длительность доставки: 10 мс – 1 с.

3. Произвольная смесь 1 и 2 класса.

Класс 1 соответствует трафику управления и навигации в бортовых сетях. Класс 2 соответствует “медиа”-трафику, такому как изображения с метеорологического радара. Класс 3 представляет

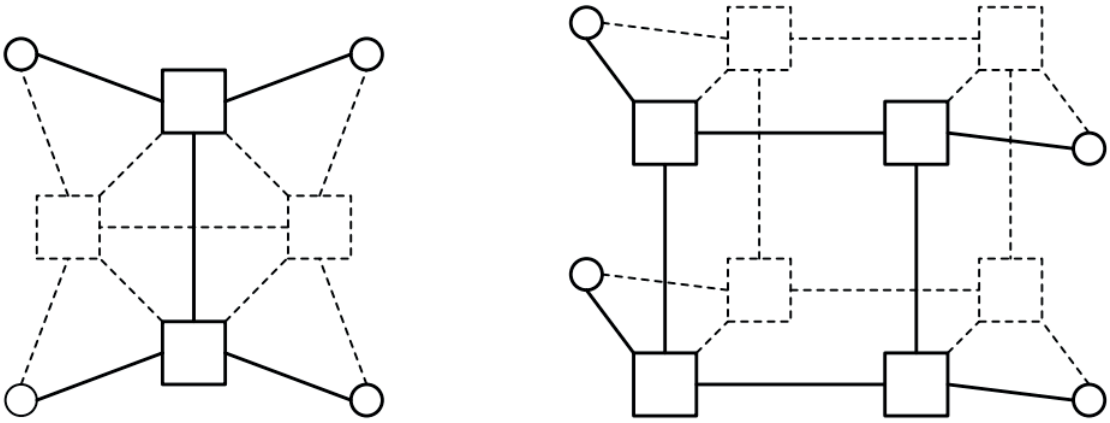

Рис. 3. Первая и вторая пара (базовая; максимальная) сетевых топологий.

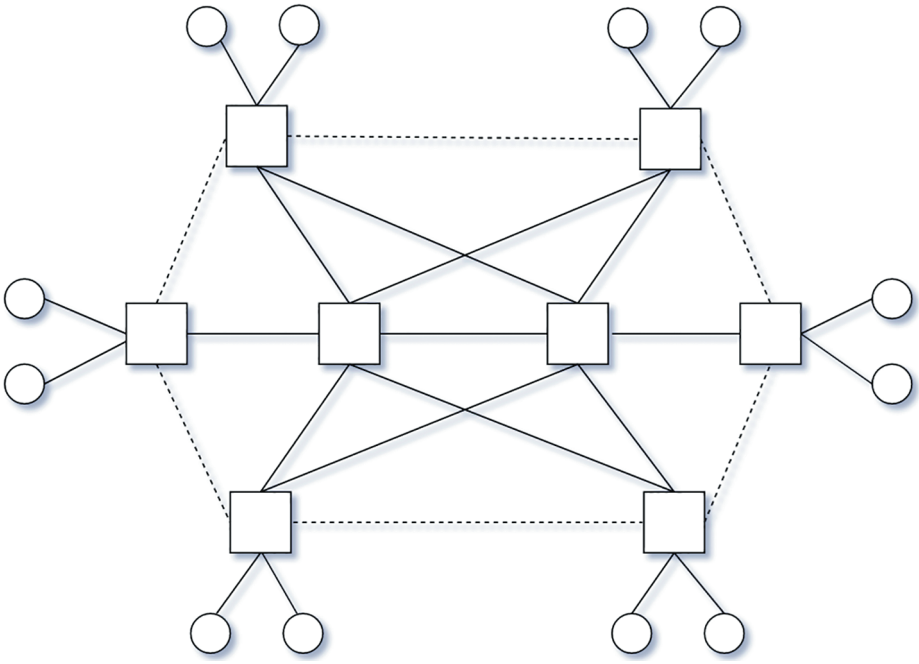

Рис. 4. Третья пара (базовая; максимальная) сетевых топологий.

гетерогенный трафик. Для каждого типа набора сообщений параметры были выбраны случайным образом в указанных диапазонах.

Для каждой пары (топология сети, тип набора сообщений) было сгенерировано и использовано в экспериментах 10 наборов сообщений. В зависимости от типа набора сообщений и конфигурации сети набор сообщений включал от 100 до 300 периодических сообщений, что согласуется с имеющимися данными о характеристиках трафика в бортовых сетях [14].

Эксперименты проводились с параметрами алгоритма: $m = 2$, $h = 4$, $p = 3$. Параметр m были выбран так, чтобы алгоритм не превращался в переборный. Параметр h был выбран таким из эвристических соображений, чтобы рассматривались достаточно “различные” маршруты передачи. Параметр p также был выбран из эвристических соображений, чтобы процедура рассматривала как максимум три различных маршрута.

Результаты экспериментального исследования представлены на рисунке 5. На рисунке видно, что в большинстве случаев для 100% сообщений успешно строится расписание передач. Максимальная сеть в большинстве случаев также успешно упрощается до базовой сети. Даже для третьей топологии, которая имеет большое количество альтернативных маршрутов для передачи

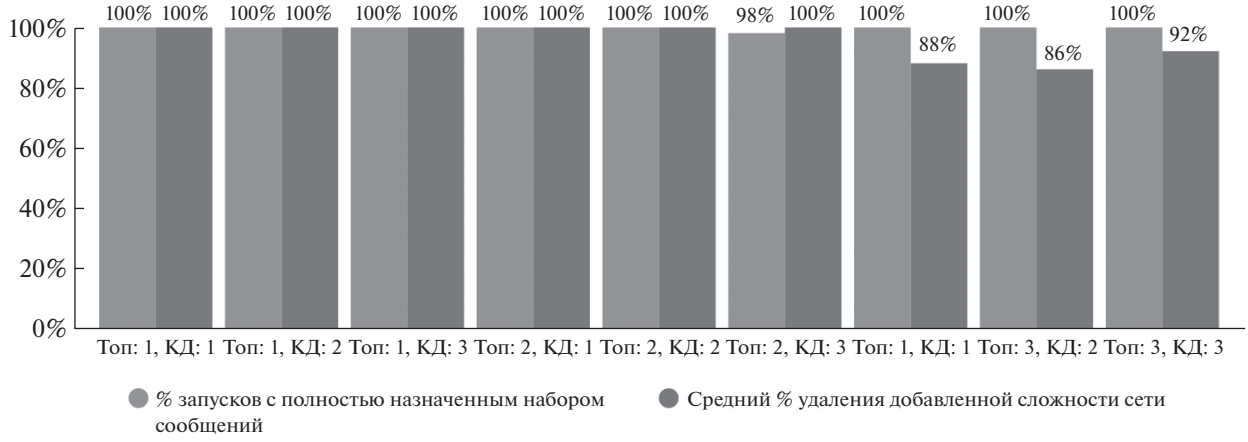

Рис. 5. Результаты экспериментального исследования для TSN сетей. (Топ:N – топология N. КД:K – класс данных K).

каждого сообщения, имеем более чем 85% удаление избыточной сложности сети.

Для 2 топологии и 3 класса данных имеем 2% запусков, где не удалось назначить полностью набор сообщений, что обуславливается тем, что алгоритм является жадным и процедуры ограниченного перебора не всегда помогают получить точное решение за оптимальное время.

6. ЗАКЛЮЧЕНИЕ

Предложенный в работе алгоритм для исходно заданного множества периодических сообщений осуществляет построение коммутируемой сети обмена с временной синхронизацией минимальной сложности необходимой для передачи сообщений в реальном времени и строит систему расписаний передач.

Результаты экспериментального исследования показали, что представленный алгоритм успешно удаляет избыточную сложность сети (не менее 85% сокращение избыточности).

СПИСОК ЛИТЕРАТУРЫ

1. ARINC Specification 651. Design Guidance for Integrated Modular Avionics. Airlines Electronic Engineering Committee, 1997.
2. ARINC Specification 653. Avionics Application Software Standard Interface. Airlines Electronic Engineering Committee, 2007.
3. INCITS 373. Information Technology – Fibre Channel Framing and Signaling Interface (FC-FS). International Committee for Information Technology Standards, 2003.
4. ARINC Specification 664. Aircraft Data Network, Part 7. Avionics Full Duplex Switched Ethernet (AFDX) Network. Airlines Electronic Engineering Committee, 2005.
5. IEEE 802.1Q – Local and Metropolitan Area Networks – Bridges and Bridged Networks. Institute of Electrical and Electronics Engineers, 2018.
6. IEEE 802.1BA-2011 – IEEE Standard for Local and metropolitan area networks – Audio Video Bridging (AVB) Systems. Institute of Electrical and Electronics Engineers, 2011.
7. *D. Bruckner et al.* An Introduction to OPC UA TSN for Industrial Communication Systems. in Proceedings of the IEEE. 2019. V. 107. № 6.
8. *Lee J., Park S.* Time-Sensitive Network (TSN) Experiment in Sensor-Based Integrated Environment for Autonomous Driving. Sensors, 2019.
9. *Lin Zhao, Feng He, Ershuai Li, Jun Lu.* Comparison of Time Sensitive Networking (TSN) and TTEthernet. IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), 2018.
10. *Костенко В.А.* Архитектура программно-аппаратных комплексов бортового оборудования. Изв. вузов. Приборостроение. 2017. Т. 60. № 3.
11. *Al-Kuwaiti M., Kyriakopoulos N., Hussein S.* A comparative analysis of network dependability, fault-tolerance, reliability, security, and survivability. IEEE Communications Surveys & Tutrials. 2009. V. 11. № 2.
12. *Zhao L., Pop P., Zheng Z., Li Q.* Timing Analysis of AVB Traffic in TSN Networks Using Network Calculus. 2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), 2018.
13. *Sune Mølgaard Laursen, P. Pop, W. Steiner.* Routing optimization of AVB streams in TSN networks. ACM SIGBED Review, 2016.
14. *Amari A., Mifdaoui A.* Specification and Performance Indicators of AeroRing – A Multiple-Ring Ethernet Network for Avionics Embedded Systems. Sensors. 2018. V. 18.

EDN: NRYCKR
ПРОГРАММИРОВАНИЕ, 2022, № 6, с. 54–58

ТЕОРИЯ ПРОГРАММИРОВАНИЯ: ФОРМАЛЬНЫЕ МОДЕЛИ И СЕМАНТИКА

УДК 004.891.2

АЛГОРИТМ ХЕШИРОВАНИЯ ИЗОБРАЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ

© 2022 г. О. В. Куликова[a],* (https://orcid.org/ 0000-0002-9526-3078),
Г. С. Домбаян[a],(https://orcid.org/ 0000-0002-8890-288X)**

[a] Донской государственный технический университет
344002 Ростов-на-Дону, пл. Гагарина, 1, Россия

**E-mail: kov0768@list.ru*

***E-mail: nuken_96@mail.ru*

Поступила в редакцию 03.03.2022 г.
После доработки 24.06.2022 г.
Принята к публикации 27.06.2022 г.

Цель исследования заключается в разработке алгоритма для хеширования изображений с использованием сверточных нейронных сетей. Алгоритм, предложенный в данной работе, реализуется в три этапа:

1) предварительное обучение нейронной сети на тренировочных данных;

2) настройка нейронной сети для одновременного обучения нейронной сети семантическим признакам изображения и аппроксимирующей хеш-подобной функции для вычисления хеш-кодов;

3) извлечение изображений с помощью предложенного алгоритма иерархического глубокого поиска.

DOI: 10.31857/S0132347422060061

1. ВВЕДЕНИЕ

Хеширование применяется в самых разнообразных областях использования информационных технологий: в базах данных для ускорения поиска по ключу, в криптосистемах, в высокоуровневых языках программирования для реализации разнообразных структур данных и алгоритмов и пр. [1]. В информационной безопасности хеширование становится краеугольным камнем таких задач, как: хранение паролей, создание уникальных криптографических ключей и электронной цифровой подписи, аудит подлинности и целостности документов в ПК. Хеширование и в настоящее время остается достаточно популярной и востребованной областью информационных технологий, а благодаря достижениям в индустрии микротранзисторов и ростом вычислительных мощностей появилась возможность модификации и возможного улучшения алгоритмов хеширования. С ростом общего объема визуальной информации, передаваемой по сети, возросла необходимость в эффективном хранении и поиске изображений в больших базах данных. Для решения данных проблем идеально подходит хеширование изображений с сохранением их семантических признаков. Данная область научного исследования сравнительно молода и развивается всего на протяжении последних десяти лет, поэтому требует дальнейших усовершенствований и модификаций используемых алгоритмов.

2. СРАВНИТЕЛЬНЫЙ АНАЛИЗ ГЛУБОКИХ МЕТОДОВ ХЕШИРОВАНИЯ ИЗОБРАЖЕНИЙ

Подробный сравнительный анализ эффективности применения традиционных методов машинного обучения и сверочных нейронных сетей для решения задачи семантического хеширования представлен в работах [2] и [3]. В таблице 1 представлена в порядке возрастания точность методов глубокого хеширования с 12/16, 32 и 48-битными хеш-значениями. Стоит отметить, что несмотря на высокую точность предложенных в работах [2] и [3] моделях и эффективность алгоритма глубокого иерархического поиска, данные модели предъявляют высокие требования к вычислительным ресурсам обучаемой машины, а это нежелательный фактор при необходимости переобучения использованной модели под конкретные технические спецификации. Также некоторые из рассмотренных моделей дают большую погрешность в извлечении похожих картинок по хеш-кодам.

Как видно, наиболее эффективным является метод глубокого обучения на бинарных хеш-значениях, в связи с чем можно определить следую-

Таблица 1. Сравнительный анализ точности методов глубокого хеширования

Применяемый метод хеширования	Точность метода с 12/16-битным хеш-значением	Точность метода с 32-битным хеш-значением	Точность метода с 48-битным хеш-значением
Неуправляемое глубокое хеширование	19.43%	24.86%	23.95%
Бинарное глубокое обучение	67.32%	69.63%	66.45%
Глубокое попарное обучение	71.3%	74.4%	75.7%
Глубокое обучение на бинарных хеш-значениях	89.3%	89.72%	89.73%

щие требования к искомому хеш-значению для изображения:

– сохранение максимально возможного количества информации из входного изображения;

– минимальные потери информации из входного изображения при извлечении главных признаков;

– инвариантность относительно афинных преобразований [4] и аугментации входных данных [5];

– равномерное распределение битов хеш-значения для увеличения полезной информации.

3. ПРЕДВАРИТЕЛЬНОЕ ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ НА ТРЕНИРОВОЧНЫХ ДАННЫХ

Для реализации сверточной нейронной сети была выбрана архитектура AlexNet, поскольку не требует больших вычислительных ресурсов для обучения или переобучения сети на пользовательском наборе изображений.

На рис. 1 подробно представлена структура модели. Данная модель была предобучена на базе данных ImageNet, предназначенной для отработки и тестирования методов распознавания образов и машинного зрения. Следует отметить, что модель была модифицирована для выполнения поставленной задачи.

Модель включает в себя пять сверточных слоев, предназначенных для минимизации размерности входного изображения, и три полносвязных слоя, которые занимаются собственной классификацией. Был добавлен еще один слой между седьмым и восьмым полносвязными слоями, задачей которого является обучение хешированию изображений. Данный слой принимает на вход семантические признаки изображения, полученные на предыдущем слое и затем вычисляет хеш-значения изображения.

CONV1 – первый сверточный слой
MAX POOL1 – первый pooling-слой
NORM1 – первый нормализация
CONV2 – второй сверточный слой
MAX POOL2 – второй pooling-слой
NORM2 – вторая нормализация
CONV3 – третий сверточный слой
CONV4 – четвертый сверточный слой
CONV5 – пятый сверточный слой
Max POOL3 – третий pooling-слой
FC6 – полносвязный слой
FC7 – полносвязный слой
FC8 – полносвязный слой

Рис. 1. Порядок слоев в модифицированной модели AlexNet.

4. НАСТРОЙКА И ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ

Для обучения хеширующего слоя используется погрешность сверточной нейронной сети по классификации изображений. Также, для данной нейронной сети специально была подобрана функция потерь:

$$\arg\min_{W} \alpha\sum_{n=1}^{N} L(y_n, \hat{y}_n) + \lambda\|W\|^2 - \beta\sum_{n=1}^{N}\|a_n^H - 0.5\|^2 + \gamma\sum_{n=1}^{N}(mean(a_n^H) - 0.5)^2$$

Третье слагаемое данной функции служит для бинаризации выходов хеширующего слоя для составления хеш-значения. Четвертое слагаемое предназначается для равномерного распределения битов хеш-значения, чтобы оно переносило максимально возможное количество информации из входного изображения. Также, для обучения нейронной сети был использован метод регуляризации “drop-out” для уменьшения вычислительной сложности и пакетная нормализация.

Обучение СНС происходило традиционным образом с помощью градиентного спуска [6]. Схема алгоритма обучения нейронной сети представлена на рис. 2. На схеме алгоритма обучения можно видеть, что в каждой итерации (эпохе) обучения нейронная сеть выполняет прямой проход, в котором вычисляет значения функций активации для каждого слоя и сохраняет состояния весовых коэффициентов и параметров сдвига, а затем выполняет обратный проход, в котором вычисляет значения производных функций активации. С помощью полученных значений производных

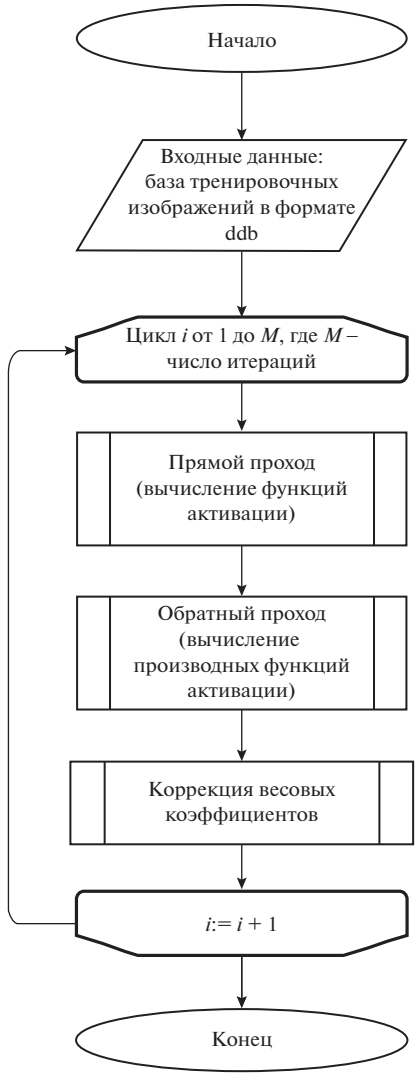

Рис. 2. Схема алгоритма обучения нейронной сети.

нейронная сеть корректирует значения весовых коэффициентов на каждом слое, тем самым минимизируя функцию ошибки.

Недавние исследования [7], [8] и [9] показали, что значения функций активации последних слоев нейронной сети могут служить визуальным дескриптором входного изображения. Использование таких визуальных дескрипторов демонстрирует впечатляющие результаты для задач классификации изображений, поиска и других задач компьютерного зрения. Однако такие визуальные дескрипторы многомерны и требуют больших вычислительных ресурсозатрат. В данной работе предлагается наряду с использованием СНС для извлечения семантических признаков изображения использовать дополнительный скрытый слой, задачей обучения которого будет аппроксимация хеш-функции для заданного вектора семантических признаков.

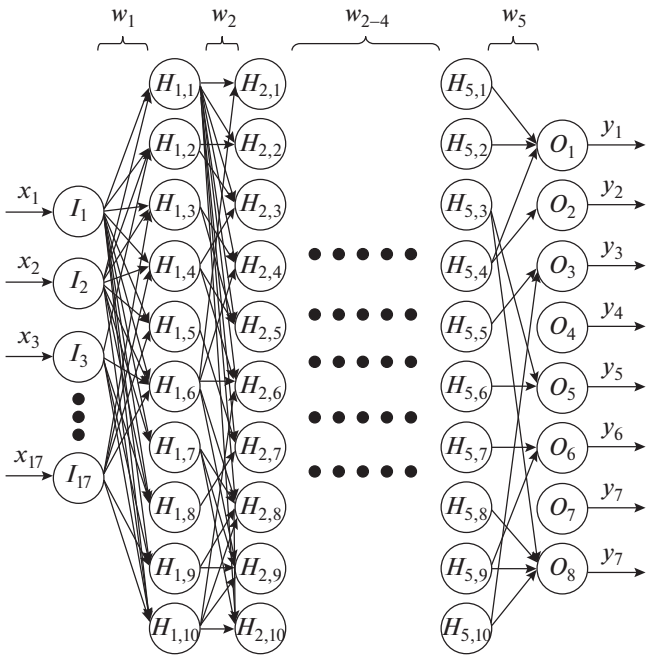

Рис. 3. Структура скрытого слоя СНС для вычисления хеш-значения изображения.

Чтобы облегчить эффективный поиск изображений, используется практичный способ преобразования результатов функций активации в двоичные коды. Такие двоичные компактные коды могут быть использованы для хеширования, а затем, для вычисления расстояния Хэмминга [10]. В данной работе предлагается наряду с использованием СНС для извлечения семантических признаков изображения использовать дополнительный скрытый слой, задачей обучения которого будет аппроксимация хеш-функции для заданного вектора семантических признаков. Обучение скрытого слоя для вычисления хеш-значения будет происходить благодаря имеющимся помеченным тренировочным данным и минимизации функции потерь от погрешности работы СНС. Структура скрытого слоя показана на рис. 3. На данном рисунке видно, что скрытый слой нейронной сети имеет параметры (17, 50, 8). Это означает, что данный слой использует для вычисления двоичных кодов 17 входных семантических признаков, 50 скрытых нейронов и 8 выходных нейронов, значения функции активации которых используются для составления хеш-значения. Указанные выше параметры скрытого слоя нейронной сети являются компромиссом между требуемой точностью извлечения изображений из базы данных и вычислительными ресурсами обучаемой машины.

Настройки модели СНС сохраняются в специальном proto-файле, структура которого пред-

```
name: "AlexNet"
layer {
  name: "data"
  type: "Input"
  top: "data"
  input_param { shape: { dim: 10 dim: 3 dim: 227 dim: 227 } }
}
layer {
  name: "conv1"
  type: "Convolution"
  bottom: "data"
  top: "conv1"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  convolution_param {
    num_output: 96
    kernel_size: 11
    stride: 4
  }
}
layer {
  name: "relu1"
  type: "ReLU"
  bottom: "conv1"
  top: "conv1"
}
```

Рис. 4. Архитектура СНС для хеширования изображений.

ставлена на рис. 4. Лаконичность такого формата позволяет эффективно создавать и сохранять новые архитектуры СНС.

5. ИЗВЛЕЧЕНИЕ ИЗОБРАЖЕНИЙ С ПОМОЩЬЮ ПРЕДЛОЖЕННОГО АЛГОРИТМА ИЕРАРХИЧЕСКОГО ГЛУБОКОГО ПОИСКА

Поиск изображения на основе его хеш-значения основан на алгоритме нахождения приблизительных ближайших соседей. Схему алгоритма поиска можно увидеть на рис. 5.

Извлечение изображения из базы данных происходит в несколько этапов:

– получение списка кандидатов с помощью алгоритма приблизительного ближайшего соседа;

– определение наилучших кандидатов с помощью минимизации расстояния Хэмминга между векторами главных признаков изображений;
– сортировка списка наилучших кандидатов в порядке возрастания расстояния Хэмминга.

Рис. 5. Схема алгоритма вычисления хеш-значения для изображения на основании вектора его семантических признаков.

На основании исследований, проведенных в работах [11], [12] и [13] был сделан вывод о том, что расстояние Хэмминга – оптимальная характеристика определения похожих изображений. Поэтому минимальное расстояние Хэмминга между хеш-значениями изображений указывает на то, что должны быть похожи при условии корректной работы СНС и метода извлечения главных признаков изображения.

6. ВЫВОДЫ

В настоящей работе было проведено алгоритмическое конструирование программного средства хеширования изображений. Также были рассмотрены следующие алгоритмы:

– бинаризация входных данных;

– инициализация нейронной сети;

– обучение нейронной сети;

– извлечение хеш-значения для входного изображения на основе его главных признаков;

– поиск изображения на основе его хеш-значения.

На основании результатов данной работы будет реализовано программное конструирование программного средства хеширования изображений.

СПИСОК ЛИТЕРАТУРЫ

1. Информационная технология. Криптографическая защита информации. Функция хеширования. ГОСТ Р 34.11-2012. Москва, Стандартинформ, 2012.
2. Методика определения угроз безопасности информации в информационных системах. Текст: электронный // Методический документ ФСТЭК России: [сайт]. 2015. https://mindstep.ru/wiki/index.php/Методика_определения_угроз_безопасности_информации_в_информационных_системах (дата обращения 26.09.2020 г.).
3. Защита от несанкционированного доступа к информации. Термины и определения: Руководящий документ ФСТЭК России [утверждено решением председателя Гостехкомиссии при Президенте Российской Федерации от 30 марта 1992 года]. Москва: Кремль, 1992. 8 с.
4. *Ершов А.В.* Линейные и аффинные пространства и отображения. М.: МФТИ, 2016. 69 с.
5. *Емельянов С.О.* Методы аугментации обучающих выборок в задачах классификации изображений / Емельянов С.О., Иванова А.А., Швец Е.А., Николаев Д.П. // Сенсорные системы. 2018. Т. 32. № 3.
6. *Городецкий С.Ю., Гришагин В.А.* Нелинейное программирование и многоэкстремальная оптимизация. Нижний Новгород: Издательство Нижегородского Университета, 2007. С. 357–363.
7. Object recognition from local scale-invariant features / David Lowe. Text: electronic // IEEE (ICCV). – 1999. – https://doi.org/10.1109/ICCV.1999.790410. http://ieeexplore.ieee.org/document/790410 (date of the application: 16.01.2021).
8. Convolutional recurrent neural networks: learning spatial dependencies for image representation / Zhen Zuo, Bing Shuai [et al.]. Text: electronic // IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVRPW). 2015. https://doi.org/10.1109/CVPRW.2015.7301268. http://ieeexplore.ieee.org/document/7301268 (date of the application: 30.01.2021). Access mode: free.
9. Exploiting local features from deep networks for image retrieval / Joe Yue-Hei Ng, Fan Yang [et al.]. Text: electronic // IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. 2015. https://doi.org/10.1109/CVPRW.2015.7301272. http://ieeexplore.ieee.org/document/7301272 (date of the application: 30.01.2021).
10. Alex X. Liu, Ke Shen, Eric Torng. Large Scale Hamming Distance Query Processing. ICDE Conference, 2011. P. 553–564.
11. Affinity CNN: learning pixel-centric pairwise relations for figure/ground embedding / Michaile Maire. – Text: electronic // IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVRP). 2016. https://doi.org/10.1109/CVPR.2016.26. http://ieeexplore.ieee.org/document/7780395 (date of the application: 14.01.2021).
12. Object contour detection with a fully convolutional encoder-decoder network / Michaile Maire, Takuya Narihira [et al.]. Text: electronic // IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVRP). 2016. https://doi.org/10.1109/CVPR.2016.28. http://ieeexplore.ieee.org/document/7780397 (date of the application: 14.01.2021).
13. Learning relaxed deep supervision for better edge detection / Yu Liu, Michael Lew [et al.]. Text: electronic // IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVRP). 2016. https://doi.org/10.1109/CVPR.2016.32. http://ieeexplore.ieee.org/document/7780401 (date of the application: 14.01.2021).