# СОДЕРЖАНИЕ

**Номер 4, 2021**

## ТЕОРИЯ ПРОГРАММИРОВАНИЯ: ФОРМАЛЬНЫЕ МОДЕЛИ И СЕМАНТИКА



## ПАРАЛЛЕЛЬНОЕ И РАСПРЕДЕЛЕННОЕ ПРОГРАММИРОВАНИЕ



## ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ



## ИНФОРМАЦИОННЫЙ ПОИСК



## КОМПЬЮТЕРНАЯ ГРАФИКА И ВИЗУАЛИЗАЦИЯ

# CONTENTS

**No. 4, 2021**

*ПРОГРАММИРОВАНИЕ, 2021, № 4, с. 3–13*

## ТЕОРИЯ ПРОГРАММИРОВАНИЯ: ФОРМАЛЬНЫЕ МОДЕЛИ И СЕМАНТИКА

*УДК 510.22*

# p-АДИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ ПОДМНОЖЕСТВ ОГРАНИЧЕННОГО ЧИСЛОВОГО МНОЖЕСТВА

**© 2021 г. В. П. Бочарников[a],*, С. В. Свешников[a],****

[a] *Консалтинговая группа ИНЭКС-FT 03011 Киев, ул. Десятинная 13а, Украина*

**E-mail: bocharnikovvp@gmail.com*

***E-mail: sv367@ukr.net*

Поступила в редакцию 18.01.2021 г.
После доработки 03.02.2021 г.
Принята к публикации 04.03.2021 г.

В статье показано, что кольцо целых p-адических чисел $Z_p$ может быть использовано для представления подмножеств ограниченного числового множества. Предложен подход к определению множества p-адических шаров, объединением образов которых является заданное подмножество ограниченного числового множества. Даны определения покрытия множества p-адических шаров и p-адической плотности подмножества ограниченного числового множества. Заданы операции p-адического пересечения, объединения и дополнения над множествами p-адических шаров, которые могут задавать соответствующую алгебру.

**DOI:** 10.31857/S0132347421040026

## 1. ВВЕДЕНИЕ

В ряде задач, например таких как исследования энергетических состояний кристаллической структуры (которая определяется электромагнитным взаимодействием ядер и электронов составляющих ее атомов) [1], энергетических ландшафтов движения мешкообразных структур везикул, которые перемещают гормоны и нейротрансмиттеры (например, инсулин и серотонин) по клеткам и телу [2], ландшафта потенциальной энергии состояния белковой молекулы [3], энергетического ландшафта скалярного поля вакуума [4] и других задач необходимо моделирование энергетических ландшафтов. Данные ландшафты представляются в виде скалярных полей, где каждой точке пространства (как правило, это пространство $R^n$, где $R$ множество действительных чисел), ставится в соответствие скалярная величина, например значения энергии в данной точке пространства, то есть, чаще всего задается скалярная функция $\nu(\cdot)\colon R^n \to R$.

Для практического моделирования сложных энергетических ландшафтов был предложен эффективный подход на основе формализации расположения энергетических бассейнов [5]. При этом множество состояний описывалось набором квазиравновесных состояний (локальных минимумов), которые при помощи эквипотенциальных сечений объединялись в “бассейны” минимумов, иерархически вложенных друг в друга. Иерархия сечений множества квазиравновесных состояний, разделенных энергетическими барьерами на ландшафте, задает иерархическую структуру бассейнов [6]. В этом случае целесообразно, чтобы область определения функции скалярного поля учитывала потенциальную иерархическую структуру поля.

Для описания межбассейновой динамики энергетический ландшафт считается фиксированным на время исследования. Однако, сам энергетический ландшафт может изменяться. Для описания динамики изменения энергетического ландшафта необходимо иметь возможность на ландшафте выделить произвольное подмножество, которое может изменить свой энергетический уровень. Следовательно, должна быть возможность формально задать данное подмножество учитывая потенциальную иерархическую структуру энергетического ландшафта. Для учета наложения ряда изменений ландшафта необходимо иметь возможность выполнения логических операций над такими подмножествами, которые образуют соответствующую алгебру подмножеств.

## 2. ПОСТАНОВКА ЗАДАЧИ

Пример 1. Рассмотрим ландшафт в виде простейшего одномерного скалярного поля, заданного на интервале $I = [0,1] \subset R$ с двумя энергетическими бассейнами (рис. 1а), где $\nu_A(\cdot), \nu_B(\cdot)$ – скалярные величины энергий бассейнов $A, B \subseteq I$,

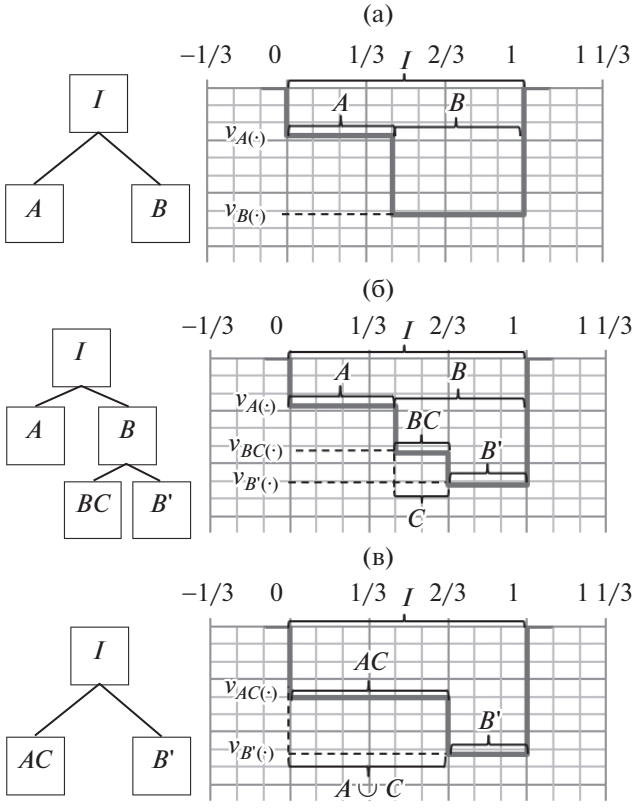


**Рис. 1.** Изменение одномерного скалярного поля.

соответственно. Данный ландшафт представляется в виде графа [5], изображенного на данном рисунке. Изменение ландшафта происходит под воздействием внешних факторов. Пусть в первый момент изменения происходят на подмножестве $C \subseteq I$ с энергией $\nu_C(\cdot)$. В результате в области $BC = C \cap B$ возникает новый бассейн с энергией $\nu_{BC}(\cdot) = F(\nu_C(\cdot), \nu_B(\cdot))$ (рис. 1б), где $F(\cdot)$ некоторая агрегирующая функция. При этом соответствующий граф ландшафта изменится. Во второй момент, допустим, что внешние факторы, действующие в области $A \cup C$ приводят к единому бассейну $AC$ с энергией $\nu_{AC}(\cdot) = F(\nu_C(\cdot), \nu_A(\cdot))$ и соответственно изменению графа ландшафта (рис. 1в). Из данного простейшего примера можно сделать три замечания. Во-первых, для определения нового распределения скалярного поля $\nu(\cdot)$ необходимо иметь возможность определения всех образовавшихся подмножеств (бассейнов) интервала $I$, которые формируются на основе алгебры подмножеств. Во-вторых, бассейны образуют иерархические структуры, которые можно представить соответствующим графом, и это необходимо учитывать при изменении скалярного поля ландшафта, “выделении” и “поглощении” отдельных бассейнов. В-третьих, при описании ландшафта, используя иерархическую структуру расположения бассейнов, можно регулировать уровень его детализации.

Как было показано в [8] координаты точек $b$ в которых реально может быть измерено скалярное поле принадлежат множеству рациональных чисел $Q$. Для нашего примера $b \in X \subset Q$, где $X$ ограничено отрезком $[0,1] \subset R$, $R$ множество вещественных чисел со стандартной метрикой. Для каждого $b \in X$ справедливо разложение $b = p^k \cdot \frac{m}{n}$, где $p$ – выбранное простое число, $k, m, n \in Z$, а $\frac{m}{n}$ – несократимая дробь, где $m$ и $n$ не делятся на $p$ [7]. Согласно теореме Островского [8], описывающей все возможные нормы, поле рациональных чисел может быть пополнено либо по евклидовой норме, либо по неархимедовой p-адической норме вида $|b|_p = p^{-k}$, которая удовлетворяет условию сильного неравенства треугольника:

$$|b + c|_p \le \max(|b|_p, |c|_p).$$

Пополнение $Q$ по p-адической норме $|b|_p$ приводит к полю p-адических чисел $Q_p$, которое естественным образом может отражать иерархическую структуру [9]. Любое p-адическое число $r \in Q_p$ отличное от нуля имеет вид [10]:

$$r = \sum_{l=-m}^{+\infty} q_l \cdot p^l, \quad q_l = 0, \ldots, p-1, q_{-m} \neq 0, \quad m \in Z.$$

Каноническая запись p-адического числа будет $r = (\ldots q_l q_{l-1} \ldots q_1, q_0 q_{-1} \ldots q_{-m})_p$, то есть бесконечная влево и конечная вправо последовательность целых чисел $q_l = 0, \ldots, p-1$. p-адические числа с нормой $|r|_p \le 1$, для которых $m \ge 0$ образуют кольцо целых p-адических чисел $Z_p$ [11]. Их каноническая запись имеет вид бесконечной влево последовательности $r = (\ldots q_l q_{l-1} \ldots q_1 q_0)_p$ целых чисел $q_l$. Иногда для записи канонической формы целых p-адических чисел для удобства используют бесконечную вправо последовательность целых чисел $q_l$ в виде $r = (q_0 q_1 \ldots q_l \ldots)_p$ [12], которую мы в дальнейшем будем использовать.

Для множества p-адических чисел $Q_p$ существует непрерывное отображение вида $\theta(r) : Q_p \to R_+$, где $R_+$ – множество неотрицательных действительных чисел [11]:

$$\theta(r) = \sum_{l=m}^{+\infty} q_l \cdot p^{-l-1}, \quad q_l = 0, \ldots, p-1, \quad m \in Z.$$

Отображение $\theta(r)$ сюръективно, взаимно однозначно почти всюду, то есть сохраняет меру (переводит p-адическую меру Хаара в меру Лебега на полупрямой), непрерывно и гёльдерово с по-

казателем 1 [13]. При этом непересекающиеся шары отображаются на интервалы, которые не пересекаются или имеют пересечение нулевой меры. Образ целого p-адического числа $r \in Z_p$ в единичном интервале вещественных чисел будет определяться соотношением вида [14]:

$$\varphi(r) = p \cdot \theta(r) = \sum_{l=0}^{+\infty} q_l \cdot p^{-l}, \quad q_l = 0,\ldots, p-1.$$

Таким образом, каждая точка $x \in I \subset R$ в которой задано одномерное скалярное поле может рассматриваться как образ p-адической координаты $r \in Z_p$, а произвольное подмножество $A \subseteq I$ как объединение образов p-адических шаров. В этом случае, например одномерное скалярное поле (см. Пример 1), представляющее ландшафт, может быть рассмотрено как отображение вида $\nu(\cdot): Z_p \to R$. Тогда для описания изменения скалярного поля ландшафта с бассейнами необходимо иметь возможность описать произвольное подмножество $A \subseteq I$ в виде множества p-адических шаров, объединение образов которых определяют данное подмножество, а также определяет операции объединения, пересечения и отрицания на множествах p-адических шаров с учетом иерархической структуры их расположения для определения изменения поля.

## 3. ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ

*Представление подмножества ограниченного числового множества в виде образа множества p-адических шаров.* Поле p-адических чисел $Q_p$ с p-адической нормой $|\cdot|_p$ является полным метрическим пространством с ультраметрикой $\forall r_i, r_j \in Q_p$, $\rho(r_i, r_j) = |r_i - r_j|_p$, удовлетворяющей условию:

$$\rho(r_i, r_j) \le \max\{\rho(r_i, r_k), \rho(r_k, r_j)\},$$
$$\forall r_i, r_j, r_k \in Q_p.$$

Для p-адических чисел $Q_p$ ультраметрика $\rho(r_i, r_j)$ может быть определена как обобщенная метрика Кантора [15]: $\rho(r_i, r_j) = \left(\frac{1}{p}\right)^{LCP(r_i, r_j)}$, $LCP(r_i, r_j)$ – длина общего префикса последовательностей для канонического представления p-адических чисел $r_i,\ r_j \in Q_p$. В частности, для $r_i, r_j \in Z_p$ величина $LCP(r_i, r_j)$ определяется из условия:

$\forall r_i, r_j$, где $r_i = (q_0, q_1, \ldots q_l \ldots)_p$, $r_j = (a_0, a_1, \ldots a_l \ldots)_p$, $LCP(r_i, r_j) = k$, если выполняется $q_l = a_l$, $l = \overline{0,k}$, $q_{k+1} \ne a_{k+1}$.

p-Адический шар с центром $a \in I$ и радиусом $\varepsilon \in R_+$ задается выражением:

$$U_\varepsilon(a) = \{r \in Q_p \,|\, \rho(r, a) \le \varepsilon\}.$$

Известно (см. например, [10]), что p-адические шары имеют следующие свойства:

1. Пусть $U$ и $V$ два p-адических шара в $X$. Тогда мы имеем только два случая:

1a. шары упорядочены по включению (или $U \subset V$, или $V \subset U$);

1b. не пересекаются ($U \cap V = \varnothing$).

2. Каждая точка шара является его центром.

3. Каждый шар в $X$ одновременно открыт и замкнут.

Для однозначного задания p-адического шара необходимо задать его центр и радиус. Центры p-адических шаров, объединение образов которых определяет подмножество $A \subseteq I$ находится на основе алгоритма.

1. Рассматриваем ограниченное числовое множество $I \subset R$ и в нем подмножество $A \subseteq I$. Фиксируем простое число $p$ (например, $p = 3$, рис. 2). Определяем максимальную детализацию разбиения множества $I$ на подмножества как максимальное количество уровней иерархии $L$.

2. Выполняем последовательное разбиение множества $I$ на подмножества $E_{q_0,\ldots,q_l}$, $q_l = 0, \ldots, p-1$, $l = \overline{1, L}$, $q_0 = 0$ так, чтобы выполнялись условия:

$$\bigcup_{q_l=0}^{p-1} E_{q_0,\ldots,q_l} = E_{q_0,\ldots q_{l-1}}, \quad E_{q_0} = I,$$
$$\forall i_l, j_l = 0,\ldots, p-1,$$
$$E_{q_0,\ldots,i_l} \cap E_{q_0,\ldots,j_l} = \varnothing,$$
$$Card(E_{q_0,\ldots,i_l}) = Card(E_{q_0,\ldots,j_l}).$$

3. Задаем $l = 0$, фиксируем начальный индекс $q_0 = 0$ для формируемых последовательностей индексов.

4. Определяем $l = l + 1$ и выполняем проверку не пустоты пересечения подмножеств $E_{q_0,\ldots,q_l}$ с подмножеством $A \subseteq I$ для формирования последовательностей из чисел $q_l = 0,\ldots, p-1$. При этом:

4a. если $E_{q_0,\ldots,q_l} \cap A = \varnothing$, то подмножество $E_{q_0,\ldots,q_l}$ не рассматривается и последовательность не формируется;

4b. если $E_{q_0,\ldots,q_l} \cap A \ne \varnothing, E_{q_0,\ldots,q_l} \not\subseteq A$, то в последовательности фиксируем $q_l$;

4c. если $E_{q_0,\ldots,q_l} \subseteq A$, то фиксируем $q_l$ и принимаем $\forall i = 1,\ldots, +\infty, q_{l+i} = p-1$. Формирование последовательности “останавливается”.

5. Для каждого случая b) повторяем выполнение пункта 4 и формируем последовательности $(q_0 q_1 \ldots q_l q_{l+1})$.

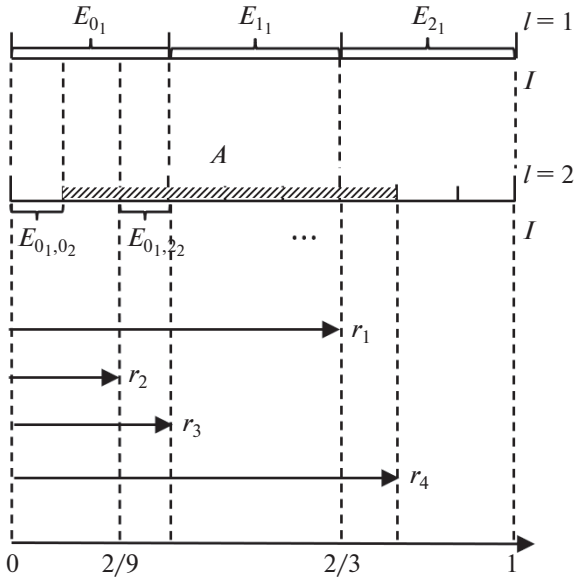


**Рис. 2.** Разбиение множества.

6. Пункты 4 и 5 выполняются до момента, пока формирование всех последовательностей не будет "остановлено" или пока не выполнится $l = L$. Для всех "неостановленных" последовательностей при $l = L$ принимаем $\forall i = 1,\ldots,+\infty, q_{L+i} = p - 1$.

Полученные с использованием приведенного алгоритма последовательности при разбиении на $p$ подмножеств, где $p$ – простое число, на каждом уровне иерархии задают положительные целые p-адические числа $r \in Z_p$ в канонической форме.

Пример 2. Применение описанного алгоритма для подмножества $A \subseteq I$ (рис. 2) формирует четыре последовательности вида $(q_0 q_1 \ldots q_l)$. Данные последовательности задают целые p-адические числа:

$$(0_0, 1_1) \vdash r_1 = (01222\ldots)_3, \quad \varphi(r_1) = \frac{2}{3};$$

$$(0_0, 0_1, 1_2) \vdash r_2 = (00122\ldots)_3, \quad \varphi(r_2) = \frac{2}{9};$$

$$(0_0, 0_1, 2_2) \vdash r_3 = (00222\ldots)_3, \quad \varphi(r_3) = \frac{1}{3};$$

$$(0_0, 2_1, 0_2) \vdash r_4 = (02022\ldots)_3, \quad \varphi(r_4) = \frac{7}{9}.$$

В соответствии с приведенным алгоритмом последовательность индексов подмножества $E_{q_0,\ldots,q_l} \subseteq A \subseteq I$ однозначно определяет p-адическое число $r$. Это p-адическое число имеет единственный образ $\varphi(r)$ во множестве $I$. Прообразом интервала $E_{q_0,\ldots,q_l} \subseteq I$ во множестве $Q_p$ будет p-адический шар [13] $U_\varepsilon(r)$ с центром $r \in I$ и радиусом $\varepsilon \in R_+$ для которого:

a. Координата центра p-адического шара для подмножества $E_{q_0,\ldots,q_l} \subseteq A$ задается p-адическим числом $r = (q_0 \ldots q_l (p-1)\ldots)_p$, которое получено на основании описанного выше алгоритма.

b. Радиус шара определяется по условию включения в шар всех p-адических чисел, имеющих образ в подмножестве $E_{q_0,\ldots,q_l} \subseteq A$ и равен $\varepsilon = \rho(r_i, r_j) = p^{-l}$, где $r_i, r_j \in Q_p$ любые p-адические числа, для которых $\varphi(r_i), \varphi(r_j) \in E_{q_0,\ldots,q_l}$.

Пример 3. Для Примера 2, где $p = 3$, p-адическое расстояние заданное метрикой Кантора для $r_1, r_2 \in Z_p$ и $r_2, r_3 \in Z_p$ будет: $\rho(r_1, r_2) = \left(\frac{1}{p}\right)^1 = \frac{1}{3}$, $\rho(r_2, r_3) = \left(\frac{1}{p}\right)^2 = \frac{1}{9}$.

Образы непересекающихся p-адических шаров отображаются на интервалы вещественных чисел, которые не пересекаются или имеют пересечение нулевой меры. Тогда произвольное подмножество $A \subseteq I$ будет определяться объединением образов p-адических шаров с указанными выше центрами и радиусами.

Пример 4. Для примера на рис. 2 подмножество $A \subseteq I$ будет объединением образов p-адических шаров (рис. 3) из множества шаров:

$$U(A) = \left\{U_{\frac{1}{3}}(r_1), U_{\frac{1}{9}}(r_2), U_{\frac{1}{9}}(r_3), U_{\frac{1}{9}}(r_4)\right\}.$$

Далее будем говорить, что множество шаров $U(A) = \{U_{\varepsilon_i}(r_i) \mid i = \overline{1, N_A}\}$, описывает подмножество $A \subseteq I$. Множество $U(A)$ целесообразно упорядочить по убыванию величины радиуса $\varepsilon_i$, а для одинаковых радиусов упорядочить по возрастанию значения $\varphi(r_i)$. Исходя из свойств p-адических шаров, выполняется условие: $\forall a, b \in Q_p$, $a \in U_{\varepsilon_i}(r_i)$, $b \in U_{\varepsilon_j}(r_j)$, $\rho(a, b) = \text{const}$.

*p-адическое приближение и покрытие множества шаров.*

Определение 1. Для множества шаров $U(A) = \{U_{\varepsilon_i}(r_i) \mid i = \overline{1, N_A}\}$, описывающих $A \subseteq I$, подмножество шаров $U_{\varepsilon_{apr}}(A) \subseteq U(A)$, с радиусами $\varepsilon_i \geq \varepsilon_{apr}$, называется p-адическим приближением подмножества $A$ с погрешностью до $\varepsilon_{apr} \in [0,1]$.

Пример 5. P-адическим приближением подмножества $A$ (Пример 4) с точностью $\varepsilon_{apr} = \frac{1}{3}$ бу-

дет подмножество $A_{\frac{1}{3}} = E_{0_0,1_1}$, описывающееся шаром $U_{\frac{1}{3}}(r_1)$.

Определение 2. Покрывающим шаром или покрытием множества шаров $\mathcal{U} = \{U_{\varepsilon_i}(r_i), |i = \overline{1,N}\}$ называется шар $U_\alpha(r)$, для которого $\forall i, U_{\varepsilon_i}(r_i) \subseteq U_\alpha(r)$. В этом случае будем записывать $U_\alpha(r) = Cov_\alpha(\mathcal{U})$.

Центром покрывающего шара $U_\alpha(r)$ будем считать точку с координатой $r = (q_0 q_1 \ldots q_l (p-1)(p-1)\ldots)_p$, где $(q_0 q_1 \ldots q_l)$ – общий префикс для последовательностей канонического представления p-адических чисел координат всех центров шаров множества $\mathcal{U}$, $l \le l_{\max}$, $l_{\max} = \min\limits_{i,j=\overline{1,N}} LCP(r_i, r_j)$ – длина максимального общего префикса p-адических чисел всех центров. Радиусом покрывающего шара $U_\alpha(r)$ будет величина $\alpha = \frac{1}{p^l}$. Множество всех покрытий для множества $\mathcal{U}$ будет $CS(\mathcal{U}) = \{Cov_{\rho^{-l}}(\mathcal{U}) | l \le l_{\max}\}$. Любое подмножество $CB(\mathcal{U}) \subseteq CB(\mathcal{U})$ задает покрывающее тело множества шаров $\mathcal{U}$.

Покрывающий шар $U_{\alpha_{min}}(r)$, для которого $l = l_{\max}$, будет минимальным покрытием множества $\mathcal{U}$. Для него радиус будет соответственно $\alpha_{\min} = p^{-l_{\max}}$. При этом для множества шаров $\mathcal{U}$ выполняется условие $\forall i, \rho(r_i, r) \le \alpha_{min}$. Центр минимального покрывающего шара будет удовлетворять условию $r = \max\limits_{i=\overline{1,N}} r_i$. Минимальное покрытие для множества из одного шара $U_{\varepsilon_i}(r_i)$ будет совпадать с данным шаром. Максимальным покрытием $U_1(1)$ любого множества $\mathcal{U}$ является шар, для которого множество $I$ является его образом. Шар $U_1(1)$ имеет центр $r = (0(p-1)(p-1)\ldots)_p$ и радиус $\alpha = 1$.

Пример 6. Для примера рис. 2 покрытиями для множества шаров $\mathcal{U} = \left\{U_{\frac{1}{9}}(r_2), U_{\frac{1}{9}}(r_3)\right\}$ будут шары $U_{\alpha_1}(r_{\alpha_1})$ с центром $r_{\alpha_1} = (00222\ldots)_3$ и радиусом $\alpha_1 = \frac{1}{3}$ и $U_{\alpha_2}(r_{\alpha_2})$ с центром $r_{\alpha_1} = (02222\ldots)_3$ и радиусом $\alpha_1 = 1$. При этом шар $U_{\alpha_1}(r_{\alpha_1})$ будет минимальным покрытием множества шаров $\mathcal{U}$.

Определение 3. Два шара $U_{\varepsilon_i}(r_i)$ и $U_{\varepsilon_j}(r_j)$ называются сходными по покрытию $U_\alpha(r)$, если шар $U_\alpha(r)$ является их покрывающим шаром. Если шары $U_{\varepsilon_i}(r_i)$ и $U_{\varepsilon_j}(r_j)$ имеют радиусы $\varepsilon_i = \varepsilon_j = p^{-l}$ и сходны по минимальному покрытию $U_\alpha(r)$ с радиусом $\alpha \le p^{1-l}$, то они называются равными по покрытию. Условие равенства шаров по покрытию будем записывать как:

$$U_{\varepsilon_i}(r_i) \equiv U_{\varepsilon_j}(r_j)\{Cov|U_\alpha(r)\}.$$

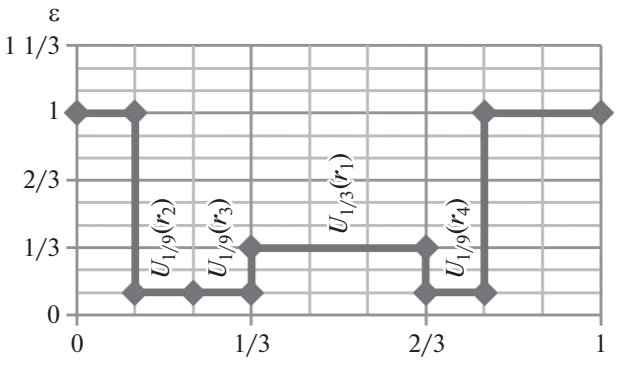


**Рис. 3.** Представление множества шарами.

В частности, для Примера 3 будем иметь $U_{\frac{1}{9}}(r_2) \equiv U_{\frac{1}{9}}(r_3)\left\{Cov\left|U_{\frac{1}{3}}(r_{\alpha_1})\right.\right\}$.

Определим во множестве шаров $\mathcal{U} = \{U_{\varepsilon_i}(r_i), |i = \overline{1,N}\}$ шар наименьшего радиуса $U_{\varepsilon'}(r') \in \mathcal{U}$, $\varepsilon' = \min\limits_i \varepsilon_i = p^{-l'}$. Пусть $U_\alpha(r), \alpha = p^{-l_\alpha}$ – покрывающий шар для данного множества шаров.

Определение 4. Высотой покрытия $U_\alpha(r)$ для множества шаров $\mathcal{U}$ называется величина, определяемая соотношением: $hgt_U(U_\alpha(r)) = l' - l_\alpha$. Если $U_\alpha(r)$ является минимальным покрытием, то $hgt_U(U_\alpha(r))$ называется минимальной высотой покрытия $U_\alpha(r)$ для множества шаров $\mathcal{U}$.

Высота покрытия шара $U_{\varepsilon_i}(r_i)$ будет $hgt_{U_{\varepsilon_i}(r_i)}(U_\alpha(r)) = l_i - l_\alpha$, где $l_i = -log_p \varepsilon_i$. Тогда для одного шара минимальная высота покрытия равна нулю.

Пример 7. Для Примера 2 наименьший радиус шара будет $\varepsilon' = \min\limits_i \varepsilon_i = p^{-2} = \frac{1}{9}$, $l' = 2$. Минимальный покрывающий шар совпадает с множеством $I$. Следовательно, $\alpha = p^0 = 1$, $l_\alpha = 0$. Тогда минимальная высота покрытия для множества шаров $U(A)$, будет определяться значением $hgt_{U(A)}(U_\alpha(r)) = l' - l_\alpha = 2 - 0 = 2$.

*p-Адическая плотность подмножества.* p-Адические шары из множества $U(A) = \{U_{\varepsilon_i}(r_i) | i = \overline{1, N_A}\}$ не

пересекаются. Мощность подмножества $A$ будет определяться в виде $Card(A) = \sum_{i=1}^{N_A} \varepsilon_i$. Для описания внутренней структуры подмножества $A \subseteq I$ определим степень его p-адической плотности. В общем случае множество плотно, если между двумя произвольными элементами множества всегда возможно найти третий элемент этого множества [16]. Если множество не плотно, то оно считается разряженным. Свойствами p-адических шаров является то, что они одновременно и открыты, и замкнуты. Поэтому использование классического определения плотности множества затруднено. Однако, интуитивно понятно, что для более плотного подмножества $A \subseteq X$ размеры "пустот" между шарами, описывающими его, должны быть минимальными.

Определение 5: Степенью p-адической плотности подмножества $A$ называется величина, которая определяется соотношением:

$$\delta(A) = \sum_{i=1}^{N_A} p^{-hgt_{U_{\varepsilon_i}(r_i)}(U_\alpha(r))} \in [0,1],$$

где $\left\{U_{\varepsilon_i}(r_i) \middle| i = \overline{1, N_A}\right\} = U(A)$ множество шаров, представляющих подмножество $A \subseteq I$, $U_\alpha(r)$ – минимальный покрывающий p-адический шар для $U(A)$.

Степень p-адической плотности показывает, насколько множество $A$ заполнено шарами на всех уровнях разбиения множества $I$. Если $\delta(A) \to 1$, то множество будем считать p-адически плотным, а при $\delta(A) \to 0$ – p-адически разряженным. Очевидно, что p-адическая плотность будет $\delta(A) = 1$, если множество $U(A)$ описывается единственным шаром, равным минимальному покрытию.

Утверждение 1. p-Адическая плотность множества $A$ будет стремиться к единице $\delta(A) \to 1$, если его мощность $Card(A)$ стремится к величине радиуса минимального покрывающего шара $U_\alpha(r)$.

Доказательство. Радиус минимального покрывающего p-адического шара $U_\alpha(r)$ определяется величиной $\alpha = p^{-l_\alpha}$. Пусть $U(A) = \{U_{\varepsilon_i}(r_i) \mid i = \overline{1, N_A}\}$, где $\forall i, \varepsilon_i = p^{-l_i}$. Подставим данные значения в формулу для p-адической плотности. Тогда имеем:

$$\delta(A) = \sum_{i=1}^{N_A} p^{-hgt_{U_{\varepsilon_i}(r_i)}(U_\alpha(r))} = \sum_{i=1}^{N_A} p^{-(l_i - l_\alpha)} =$$

$$= p^{l_\alpha} \cdot \sum_{i=1}^{N_A} p^{-l_i} = \frac{\sum_{i=1}^{N_A} p^{-l_i}}{p^{-l_\alpha}} = \frac{Card(A)}{\alpha} \to 1.$$

Тогда $\delta(A) \to 1$, если $Card(A) \to \alpha$. ■

Утверждение 2. Для множества $A$ p-адическая плотность $\delta(A) \to 0$, когда минимальный покрывающий p-адический шар $U_\alpha(r)$ имеет радиус $\alpha \neq 0$, а p-адическое приближение данного множества с погрешностью $\varepsilon_{apr} \to 0$ является пустым множеством.

Доказательство. Исходя из условия, радиус покрывающего шара $U_\alpha(r)$ будет иметь значение $\alpha = p^{-l_\alpha} \neq 0$ и, следовательно $l_\alpha \neq +\infty$. Пусть подмножество $A$ описывается шарами с радиусами $\varepsilon_i = p^{-l_i}, i = \overline{1, N_A}$. По условию утверждения p-адическое приближение при $\varepsilon_{apr} \neq 0$ будет $U_{\varepsilon_{apr}}(A) = \varnothing$. Следовательно, $\forall i, \varepsilon_i < \varepsilon_{apr}$. Но $\varepsilon_{apr} \to 0$, тогда все $\varepsilon_i \to 0$, что равносильно $l_i \to +\infty$. В этом случае $\forall i, hgt_{U_{\varepsilon_i}(r_i)}(U_\alpha(r)) = l_i - l_\alpha \to +\infty$. Тогда:

$$\delta(A) = \sum_{i=1}^{N_A} p^{-hgt_{U_{\varepsilon_i}(r_i)}(U_\alpha(r))} = \sum_{i=1}^{N_A} \frac{1}{p^{+\infty}} \to 0. \blacksquare$$

Таким образом, p-адическая плотность подмножества позволяет учесть его внутреннюю структуру. Плотность будет максимальной при совпадении подмножества $A$ с его минимальным покрытием. Минимальная плотность будет, когда шары, описывающие подмножество, имеют минимальные радиусы и максимально "разбросаны" на множестве $I$.

Пример 8. Рассмотрим подмножество $A \subseteq I$, представленное на рис. 2. Оно описывается множеством шаров $U(A) = \left\{U_{\frac{1}{3}}(r_1), U_{\frac{1}{9}}(r_2), U_{\frac{1}{9}}(r_3), U_{\frac{1}{9}}(r_4)\right\}$. Для данного множества шаров имеем $l_{i=1} = 1$, $l_{i=\overline{2,4}} = 2$. Минимальное покрытие $U_\alpha(r)$ будет иметь параметры $r_\alpha = (02222\ldots)_3$ и $\alpha = p^{-l_\alpha} = 1$. Следовательно, $l_\alpha = 0$. Высота покрытия $U_\alpha(r)$ для шаров из множества $U(A)$ будет: $hgt_{U_{\varepsilon_1}(r_1)}(U_\alpha(r)) = l_1 - l_\alpha = 1$ и $i = \overline{2,4}$, $hgt_{U_{\varepsilon_i}(r_i)}(U_\alpha(r)) = l_i - l_\alpha = 2$. Тогда степень p-адической плотности подмножества $A \subseteq I$ будет:

$$\delta(A) = p^{-hgt_{U_{\varepsilon_1}(r_1)}(U_\alpha(r))} + \sum_{i=2}^{4} p^{-hgt_{U_{\varepsilon_i}(r_i)}(U_\alpha(r))} =$$

$$= 3^{-1} + 3 \cdot 3^{-2} = \frac{2}{3}$$

*Операции над множествами, представленными p-адическими шарами.* Теоретико-множественные операции над множествами, представленными p-адическими шарами должны учитывать

свойства взаимодействия шаров. В частности учитывать, что два шара могут либо не пересекаться, либо шар с меньшим радиусом поглощается шаром с большим радиусом. Рассмотрим два шара $U_\alpha^A(r_A)$ и $U_\beta^B(r_B)$, которые относятся к множествам $A$ и $B$ соответственно. Шар $U_\alpha^A(r_A)$ будет вложенным в шар $U_\beta^B(r_B)$ если $\alpha < \beta$ и $\rho(r_A, r_B) \le \varepsilon^* = \max(\alpha, \beta)$.

Утверждение 3. Шары $U_\alpha^A(r_A)$ и $U_\beta^B(r_B)$ не пересекаются, если $\sigma(\varepsilon^*) = p^{l_{\varepsilon^*}-l} > 1$, где $l_{\varepsilon^*} = \min\limits_{i,j} LCP(r_i, r_j), \forall r_i, r_j \in U_{\varepsilon^*}(r^*) = \begin{cases} U_\alpha^A(r_A), \alpha \ge \beta, \\ U_\beta^B(r_B), \alpha < \beta, \end{cases}$ и $l = LCP(r_A, r_B)$.

Доказательство. Расстояние между центрами шаров $U_\alpha^A(r_A)$ и $U_\beta^B(r_B)$ определяется метрикой $\rho(r_A, r_B) = \left(\frac{1}{p}\right)^l$. Величина $l_{\varepsilon^*}$ задает радиус наибольшего шара $\varepsilon^* = \left(\frac{1}{p}\right)^{l_{\varepsilon^*}}$. Для того, чтобы шары не пересекались должно выполняться условие $\rho(r_A, r_B) - \varepsilon^* > 0$. После подстановки значений получим условие:

$$\left(\frac{1}{p}\right)^l - \left(\frac{1}{p}\right)^{l_{\varepsilon^*}} = \frac{p^{l_{\varepsilon^*}} - p^l}{p^l \cdot p^{l_{\varepsilon^*}}} = \frac{p^l \cdot (p^{l_{\varepsilon^*}-l} - 1)}{p^l \cdot p^{l_{\varepsilon^*}}} = \frac{(p^{l_{\varepsilon^*}-l} - 1)}{p^{l_{\varepsilon^*}}} > 0.$$

Так как рассматриваются не пустые шары, то $p^{l_{\varepsilon^*}} > 0$. Отсюда следует, что шары не будут пересекаться, если $p^{l_{\varepsilon^*}-l} = \sigma(\varepsilon^*) > 1$.■

В том случае, когда шары упорядочены по включению будет выполняться условие $\sigma(\varepsilon^*) \le 1$. В зависимости от значения $\varepsilon^*$ будет определяться поглощаемый шар. Если $\varepsilon^* = \alpha$, то поглощается шар $U_\beta^B(r_B)$, и наоборот. С учетом свойства функции $\sigma(\varepsilon^*)$ введем в рассмотрение операции поглощения и выделения p-адических шаров.

Определение 6. Операциями выделения $\otimes$ и поглощения $\oplus$ p-адических шаров называются операции, заданные бинарными функциями вида:

$$U_\alpha^A(r_A) \otimes U_\beta^B(r_B) = \begin{cases} U_\beta^B(r_B), & \sigma(\varepsilon^*) \le 1, \quad \varepsilon^* = \alpha, \\ U_\alpha^A(r_A), & \sigma(\varepsilon^*) \le 1, \quad \varepsilon^* = \beta, \\ \varnothing, & \sigma(\varepsilon^*) > 1. \end{cases}$$

$$U_\alpha^A(r_A) \oplus U_\beta^B(r_B) = \begin{cases} U_\alpha^A(r_A), & \sigma(\varepsilon^*) \le 1, \quad \varepsilon^* = \alpha, \\ U_\beta^B(r_B), & \sigma(\varepsilon^*) \le 1, \quad \varepsilon^* = \beta, \\ U_\alpha^A(r_A) \cup U_\beta^B(r_B), & \sigma(\varepsilon^*) > 1. \end{cases}$$

где $\sigma(\varepsilon^*) = p^{l_{\varepsilon^*}-l}, \varepsilon^* = \max(\alpha, \beta)$.

Результатом операции выделения для двух шаров является либо шар с меньшим радиусом, либо пустое множество, а результатом операции поглощения – либо шар с большим радиусом, либо объединение сравниваемых шаров. Если функция $\sigma(\varepsilon^*) = 1$, то $U_\alpha^A(r_A) = U_\beta^B(r_B)$. В этом случае выполняется условие:

$$U_\alpha^A(r_A) \otimes U_\beta^B(r_B) = U_\alpha^A(r_A) \cap U_\beta^B(r_B) = U_\alpha^A(r_A) \cup U_\beta^B(r_B) = U_\alpha^A(r_A) \oplus U_\beta^B(r_B).$$

Пример 9. Пусть имеем три шара $A$, $B$ и $C$ с характеристиками (рис. 4):

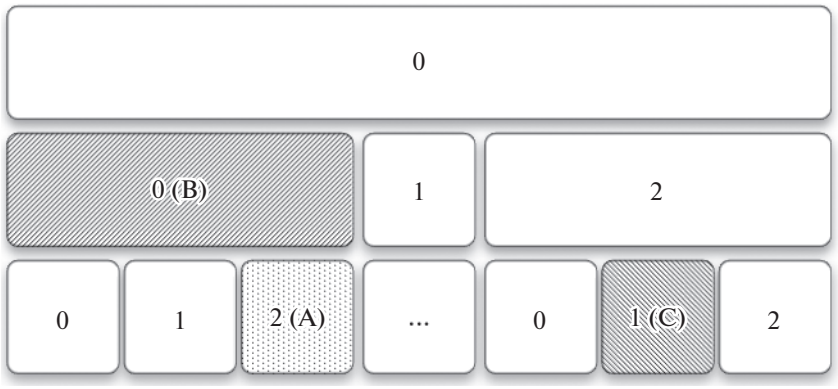


**Рис. 4.** Расположение шаров для Примера 9.

$$U^A_{\varepsilon_A}(r_A): r_A = (002222\ldots)_3; \quad \varepsilon_A = \frac{1}{9};$$
$$U^B_{\varepsilon_B}(r_B): r_B = (002222\ldots)_3; \quad \varepsilon_B = \frac{1}{3};$$
$$U^C_{\varepsilon_C}(r_C): r_C = (021222\ldots)_3; \quad \varepsilon_C = \frac{1}{9}.$$

Определим результат операции выделения для шаров $A$ и $B$. В этом случае имеем $l_{\varepsilon^*} = 1, l \to +\infty$. Тогда функция $\sigma(\varepsilon^*) < 1$. Следовательно, шары упорядочены по включению. Так как $\varepsilon_B > \varepsilon_A$, то $U^A_{\varepsilon_A}(r_A) \otimes U^B_{\varepsilon_B}(r_B) = U^A_{\varepsilon_A}(r_A)$, то есть шар $A$ выделяется на фоне шара $B$. Для шаров $A$ и $C$ рассуждения аналогичны. В этом случае $l_{\varepsilon^*} = 2, l = 0, \sigma(\varepsilon^*) = 9 > 1$. Тогда $U^A_{\varepsilon_A}(r_A) \otimes U^C_{\varepsilon_C}(r_C) = \varnothing$, то есть шары $A$ и $C$ не пересекаются. Аналогично приведенным рассуждениям операция поглощения для шаров $A$, $B$ и $C$ даст следующие результаты: $U^A_{\varepsilon_A}(r_A) \oplus U^B_{\varepsilon_B}(r_B) = U^B_{\varepsilon_B}(r_B)$, $U^A_{\varepsilon_A}(r_A) \oplus U^C_{\varepsilon_C}(r_C) = U^A_{\varepsilon_A}(r_A) \cup U^C_{\varepsilon_C}(r_C)$.

Операции выделения и поглощения позволяют определить операции p-адического пересечения и объединения множеств $A, B \subseteq I$, представленных множествами p-адических шаров.

*Операция p-адического пересечение множеств*. Рассмотрим подмножества $A, B \subseteq I$, которые описываются соответственно множествами шаров $U(A) = \{U^A_{\alpha_i}(r_i) \mid i = \overline{1, N_A}\}$, $U(B) = \{U^B_{\beta_j}(r_j) \mid j = \overline{1, N_B}\}$. Их можно представить векторами размерности $N_A, N_B$. Компонентами векторов будут соответствующие шары. Используя операцию выделения шаров, p-адическое пересечение множеств $A, B \subseteq X$ может быть представлено в виде:

$$U(A) \cap_p U(B) = \bigcup_{i,j} \{U^A_{\alpha_i}(r_i) \otimes U^B_{\beta_j}(r_j)\},$$

где операция объединения рассматривается в классическом теоретико-множественном смысле [17]. В общем случае операции $\cap_p$ и $\cap$ не совпадают, так как упорядоченные по включению шары для классической операции пересечения рассматриваются как различные элементы множеств. Операция p-адического пересечения множеств шаров $U(A)$ и $U(B)$ определяет множество всех наименьших шаров, которые были выделены. Если во множествах $U(A)$ и $U(B)$ нет упорядоченных по включению шаров, то $U(A) \cap_p U(B) = U(A) \cap U(B)$.

*Операция p-адического объединения множеств*. Рассмотрим подмножества $A, B \subseteq I$, представленные множествами шаров: $U(A) = \{U^A_{\alpha_i}(r_i) \mid i = \overline{1, N_A}\}$, $U(B) = \{U^B_{\beta_j}(r_j) \mid j = \overline{1, N_B}\}$. На основе операции поглощения p-адических шаров строится матрица с элементами $\{U^A_{\alpha_i}(r_i) \oplus U^B_{\beta_j}(r_j)\}$. Тогда соотношения $\Delta_B(U(A)) = |i(j\{U^A_{\alpha_i}(r_i) \oplus U^B_{\beta_j}(r_j)\})$ и $\Delta_A(U(B)) = |j(i\{U^A_{\alpha_i}(r_i) \oplus U^B_{\beta_j}(r_j)\})$ задают множества шаров из множеств $U(A)$ и $U(B)$, которые не были поглощены. В этом случае операция p-адического объединения множеств $A, B \subseteq X$, представленных p-адическими шарами, будет определяться соотношением:

$$U(A) \cup_p U(B) = \Delta_B(U(A)) \cup \Delta_A(U(B))$$
$$= \bigcup_{i,j} \left( \bigcap_j \{U^A_{\alpha_i}(r_i) \oplus U^B_{\beta_j}(r_j)\} \cup \bigcap_i \{U^A_{\alpha_i}(r_i) \oplus U^B_{\beta_j}(r_j)\} \right).$$

Результатом данной операции будет множество всех не поглощенных шаров, входящих хотя бы в одно из множеств $U(A)$ или $U(B)$. Если нет поглощаемых шаров, то $U(A) \cup_p U(B) = U(A) \cup U(B)$.

*Операция p-адического дополнения*. Операция p-адического дополнения должна учитывать свойства p-адических шаров. Обозначим через $U_{\alpha_l}(r_{0,q_1,\ldots,q_l})$ покрывающий шар на уровне $l$. Радиус данного шара будет $\alpha_l = \frac{1}{p^l}$. Центр этого шара определяется p-адической координатой $r_{0,q_1,\ldots,q_l} = (0q_1\ldots,q_l(p-1)(p-1)\ldots)_p$, $q_j \in \{0,(p-1)\} = QP$, $j = \overline{1,l}$. Пусть $V_{U_{\alpha_l}(r_{0,q_1,\ldots,q_l})} = \{U_{\alpha_{l+1}}(r_{0,q_1,\ldots,q_l,q_{l+1}})\}$ множество всех равных по покрытию $U_{\alpha_l}(r_{0,q_1,\ldots,q_l})$ шаров, то есть:

$$U_{\alpha_{l+1}}(r_{0,q_1,\ldots,q_l,q^n_{l+1}}) \equiv U_{\alpha_{l+1}}(r_{0,q_1,\ldots,q_l,q^m_{l+1}}) \{Cov | U_{\alpha_l}(r_{0,q_1,\ldots,q_l})\},$$
$$q^n_{l+1}, q^m_{l+1} \in QP.$$

В этом случае $\forall l = \overline{0,+\infty}, Card(V_{U_{\alpha_l}(r_{0,q_1,\ldots,q_l})}) = p$.

Пусть множество $A \subseteq I$ представляется множеством шаров $U(A) = \{U_{\varepsilon_i}(r_i) \mid i = \overline{1, N_A}\}$. p-адическое дополнение к множеству $U(A)$ во множестве $V_{U_{\alpha_l}(r_{0,q_1,\ldots,q_l})}$ будет определяться соотношением:

$$C_p[V_{U_{\alpha_l}(r_{0,q_1,\ldots,q_l})}]$$
$$= \left\{ U_{\alpha_{l+1}}(r_{0,q_1,\ldots q_l,q_{l+1}}) \mid \bigcup_{i=\overline{1,N_A}} [U_{\alpha_{l+1}}(r_{0,q_1,\ldots q_l,q_{l+1}}) \otimes U_{\varepsilon_i}(r_i)] = \varnothing \right\}.$$

Множество $C_p[V_{U_{\alpha_l}(r_{0,q_1,\ldots,q_l})}]$ содержит шары из множества $V_{U_{\alpha_l}(r_{0,q_1,\ldots,q_l})}$, которые не выделяются и не поглощаются шарами из множества $U(A)$. Тогда

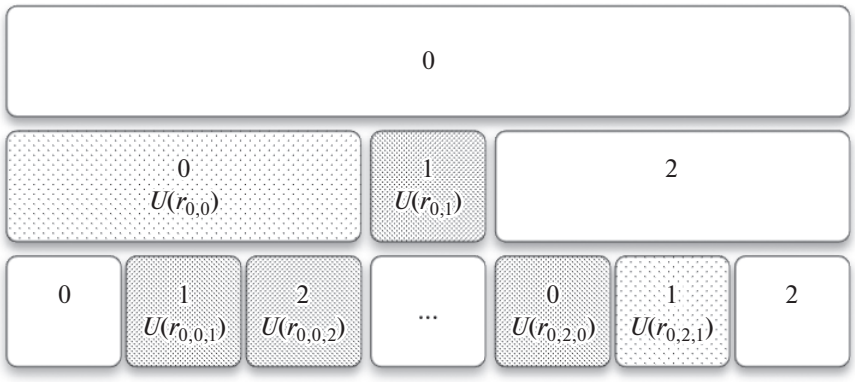


**Рис. 5.** Расположение шаров множеств $U(A)$ и $U(B)$ для Примера 10.

на уровне $l = \overline{1, L}$ все множество дополнений будет определяться выражением:

$$C_p[U^l(A)] = \bigcup_{q_1 \in QP}\left(\ldots\left(\bigcup_{q_{l-1}\in QP}\left(\bigcup_{q_l\in QP} C_p[V_{U_{\alpha_l}(r_{0,q_1,\ldots q_l})}]\right)\right)\right).$$

Рассмотрение $C_p[U^l(A)]$ целесообразно до уровня покрытий минимальных шаров из $U(A)$, то есть до $L = -\log_p(\min_i \varepsilon_i) - 1$. На уровне $l = 0$ будем иметь $C_p[U^0(A)] = C_p[V_{U_{\alpha_0}(r_0)}]$. Тогда p-адическое дополнение множества $A \subseteq X$, представленного множеством шаров $U(A)$, будет определяться соотношением:

$$C_p[U(A)] = \bigcup_{l=0,\overline{L}} C_p[U^l(A)].$$

p-Адическое дополнением является множество шаров, не входящих во множество $U(A)$ и не поглощенных шарами из множества $U(A)$. Подмножество $C_p^{apr}[U(A)] \subseteq C_p[U(A)]$ p-адического дополнения множества шаров $U(A)$ с радиусами $\varepsilon \geq \varepsilon_{apr} \in [0,1]$ будет p-адическим приближением дополнения $U(A)$ с погрешностью до $\varepsilon_{apr}$. В общем случае $C_p^{apr}[U(A)] \cup U_{\varepsilon_{apr}}(A) \subset I$. Однако максимальный шар $U_1(1)$ является минимальным покрытием ($\alpha_{\min} = 1$) объединения p-адических приближений множества шаров $U(A)$ и его дополнения:

$$U_1(1) = Cov_{\alpha_{\min}=1}(C_p^{apr}[U(A)] \cup U_{\varepsilon_{apr}}(A)).$$

Пример 10. Рассмотрим пример выполнения операций p-адического пересечения, объединения и дополнения для двух множеств $A, B \subseteq I$, представленных множествами: $U(A) = \left\{U_{\frac{1}{3}}(r_{0,1}), U_{\frac{1}{9}}(r_{0,0,1}), U_{\frac{1}{9}}(r_{0,0,2}), U_{\frac{1}{9}}(r_{0,2,0})\right\}$ и $U(B) = \left\{U_{\frac{1}{3}}(r_{0,0}), U_{\frac{1}{9}}(r_{0,2,1})\right\}$, где: $r_{0,1} = (0122\ldots)_3$, $r_{0,0,1} = (0012\ldots)_3$, $r_{0,0,2} = (0022\ldots)_3$, $r_{0,2,0} = (0202\ldots)_3$, $r_{0,0} = (0022\ldots)_3$, $r_{0,2,1} = (0212\ldots)_3$.

p-Адическое пересечение множеств $U(A)$ и $U(B)$, на основании операции выделения будет определяться соотношением:

$$U(A) \cap_p U(B) = \begin{bmatrix} \varnothing & U_{\frac{1}{9}}(r_{0,0,1}) & U_{\frac{1}{9}}(r_{0,0,2}) & \varnothing \\ \varnothing & \varnothing & \varnothing & \varnothing \end{bmatrix} = \left\{U_{\frac{1}{9}}(r_{0,0,1}), U_{\frac{1}{9}}(r_{0,0,2})\right\}.$$

Для определения p-адического объединения множеств $U(A)$ и $U(B)$, на основании операции поглощения мы сформируем матрицу вида:

$$U(A) \cup_p U(B) = \begin{bmatrix} U_{\frac{1}{3}}(r_{0,1}) \cup U_{\frac{1}{3}}(r_{0,0}) & U_{\frac{1}{3}}(r_{0,0}) & U_{\frac{1}{3}}(r_{0,0}) & U_{\frac{1}{9}}(r_{0,2,0}) \cup U_{\frac{1}{3}}(r_{0,0}) \\ U_{\frac{1}{3}}(r_{0,1}) \cup U_{\frac{1}{9}}(r_{0,2,1}) & U_{\frac{1}{9}}(r_{0,0,1}) \cup U_{\frac{1}{9}}(r_{0,2,1}) & U_{\frac{1}{9}}(r_{0,0,2}) \cup U_{\frac{1}{9}}(r_{0,2,1}) & U_{\frac{1}{9}}(r_{0,2,0}) \cup U_{\frac{1}{9}}(r_{0,2,1}) \end{bmatrix}.$$

Тогда $\Delta_B(U(A)) = \left\{U_{\frac{1}{3}}(r_{0,1}), \emptyset, \emptyset, U_{\frac{1}{9}}(r_{0,2,0})\right\}$ и $\Delta_A(U(B)) = \left\{U_{\frac{1}{3}}(r_{0,0}), U_{\frac{1}{9}}(r_{0,2,1})\right\}$. Отсюда имеем: $U(A) \cup_p U(B) = \left\{U_{\frac{1}{3}}(r_{0,0}), U_{\frac{1}{3}}(r_{0,1}), U_{\frac{1}{9}}(r_{0,2,0}), U_{\frac{1}{9}}(r_{0,2,1})\right\}$.

Для p-адического дополнения множества $A \subseteq I$ определим вспомогательные множества шаров на каждом уровне $l = \overline{0, L}$. Максимальный уровень определяется выражением $L = -\log_p(\min_i \varepsilon_i) - 1$. Тогда $L = -\log_3\left(\min_i\left\{\frac{1}{3}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}\right\}\right) - 1 = 1$, то есть $l \in \{0,1\}$.

На уровне $l = 0$ покрытие рассматривается для шаров с радиусом $\alpha_1 = \frac{1}{3}$. Дополнение на данном уровне будет $C_p[U^0(A)] = C_p[V_{U_{\alpha_0}(r_0)}]$, где

$$C_p[V_{U_{\alpha_0}(r_0)}] = \left\{U_{\alpha_1}(r_{0,q_1}) \mid \bigcup_{i=\overline{1,N_A}} [U_{\alpha_1}(r_{0,q_1}) \otimes U_{\varepsilon_i}(r_i)] = \emptyset, \right.$$
$$\left. q_1 \in \{0,1,2\}, U_{\varepsilon_i}(r_i) \in U(A)\right\} = \emptyset.$$

Для уровня $l = 1$ покрытия рассматриваются для шаров с радиусом $\alpha_2 = \frac{1}{9}$. В этом случае будем иметь: $C_p[V_{U_{\alpha_1}(r_{0,0})}] = \left\{U_{\frac{1}{9}}(r_{0,0,0})\right\}$, $C_p[V_{U_{\alpha_1}(r_{0,1})}] = \{\emptyset\}$, $C_p[V_{U_{\alpha_1}(r_{0,2})}] = \left\{U_{\frac{1}{9}}(r_{0,2,1}), U_{\frac{1}{9}}(r_{0,2,2})\right\}$. Следовательно:

$$C_p[U^1(A)] = C_p[V_{U_{\alpha_1}(r_{0,0})}] \cup C_p[V_{U_{\alpha_1}(r_{0,1})}] \cup C_p[V_{U_{\alpha_1}(r_{0,2})}]$$
$$= \left\{U_{\frac{1}{9}}(r_{0,0,0}), U_{\frac{1}{9}}(r_{0,2,1}), U_{\frac{1}{9}}(r_{0,2,2})\right\}.$$

Тогда p-адическое дополнение множества $A \subseteq I$, представленное множеством $U(A)$, будет определяться множеством шаров:

$$C_p[U(A)] = C_p[U^0(A)] \cup C_p[U^1(A)]$$
$$= \left\{U_{\frac{1}{9}}(r_{0,0,0}), U_{\frac{1}{9}}(r_{0,2,1}), U_{\frac{1}{9}}(r_{0,2,2})\right\}.$$

## 4. ВЫВОДЫ

Приведенный подход позволяет представить произвольные подмножества ограниченного числового множества $I$ в виде объединения образов множества p-адических шаров. В этом случае все множество $I$ может быть представлено как прообраз множества целых p-адических чисел $Z_p$. Предложен алгоритм определения центров p-адических шаров, которые определяются во множестве $Z_p$. Полученные p-адические операции пересечения, объединения и дополнения над множествами шаров учитывают ультраметрические свойства поля p-адических чисел. На основе данных операций может быть сформирована соответствующая алгебра множеств p-адических шаров. Максимальным элементом алгебры будет шар $U_1(1)$, имеющий образом все множество $I$. Минимальным элементом алгебры будет пустое множество шаров. Представление подмножеств ограниченного числового множества $I$ в виде подмножества p-адических шаров позволяет моделировать логику изменения нестационарных скалярных полей. Это является желательным при исследовании различных нестационарных энергетических ландшафтов. При этом множество шаров, представляющих множество $A \subseteq I$, будет определять множество нижних точек такого ландшафта, а p-адическое тело данного множества будет задавать профиль энергетического ландшафта.

## СПИСОК ЛИТЕРАТУРЫ


1. *Оганов А.Р.* USPEX: когда форма определяется содержанием. Наука из первых рук. Нарисуем – будем жить. 2012. Т. 43. № 1. С. 52–60.
2. *Хель И.* Как математик помог биологам совершить важное открытие. https://hi-news.ru/science/kak-matematik-pomog-biologam-sovershit-vazhnoe-otkrytie.html.
3. *Frauenfelder H.* The connection between low-temperature kinetics and life // Protein Structure, Molecular and Electronic Reactivity / R.H. Austin et al., eds. New York: Springer, 1987. P. 245–261.
4. *Виленкин А.* Мир многих миров. Физики в поисках иных вселенных. ООО “Издательство Астрель”, 2009. 232 с. ISBN: 978-5-271-25401-7.
5. *Becker O.M., Karplus M.* The topology of multidimensional protein energy surfaces: theory and application to peptide structure and kinetics // Journal of Chemical Physics. 1997. V. 106. P. 1495–1517.
6. *Аветисов А., Бикулов А.Х., Осипов В.А.* p-Адические модели ультразвуковой диффузии в конформационной динамике макромолекул. Труды математического института им. В.А. Стеклова, 2004. Т. 245. С. 55–64.
7. Courant R., Robbins H. What Is Mathematics? An Elementary Approach to Ideas and Methods. Oxford University Press; 2nd Edition, 1996, 592 p. ISBN-10: 0195105192
8. *Vladimirov V.S., Volovich I.V., Zelenov E.I.* P-adic Analysis and Mathematical Physics. Series on Soviet and East European Mathematics (Vol. 1). World Scientific, 1994, 340 p. ISBN 9814505765

9. Фракталы: делимость вещества как степень свободы в материаловедении: монография / А.Д. Изотов, Ф.И. Маврикиди. Самара: Изд-во Самар. гос. аэрокосм. ун-та, 2011. 128 с.: ил. ISBN 978-5-7883-0834-0
10. Katok S. p-Adic Analysis Compared with Real. Student mathematical library (V. 37), American Mathematical Society. American Mathematical Soc., 2007. 152 p. ISBN 9780821842201
11. *Волович И.В., Козырев С.В.* p-Адическая математическая физика: основные конструкции, применения к сложным и наноскопическим системам. Математическая физика и ее приложения. Вводные курсы. Выпуск 1, Самарский гос. ун-т, Самара, 2009
http://www.mi.ras.ru/noc/irreversibility/p-adicMF1.pdf
12. *Хренников А.Ю.* Моделирование процессов мышления в p-адических системах координат. М.: ФИЗМАТЛИТ, 2004. 296 с. ISBN 5-9221-0501-9
13. *Kozyrev S.V.* Wavelet theory as p-adic spectral analysis. Izv. RAN. Ser. Mat., 2002. V. 66. № 2. P. 149–158.
14. *Кононюк А.Е.* Обобщенная теория моделирования: Книга 2: Числа: количественные оценки параметров модели. Киев: “Освіта України”, 2012. 548 с. ISBN 978-966-7599-50-8
15. Deza M-M, Deza E. Encyclopedia of distances. Berlin, Springer, 2008. 412 p. (Russ. ed.: Deza M-M, Deza E. Entsiklopedicheskii slovar' rasstoyanii. Moscow, Nauka Publ., 444 p).
16. *Веселовская А.З., Шепелявая Р.Б.* Математика: логика, множества, отображения. Избранные аспекты в элементарном изложении. Изд. 2 перераб. и доп. СПб.: Изд-во С.-Петерб. ун-та, 2014. 152 с. ISBN 978-5-278-05599-7
17. *Robert R.* Stoll. Set theory and logic. Dover Publications. NewYork. 1979. 474 p. ISBN-10: 0-486-63829-4

*ПРОГРАММИРОВАНИЕ, 2021, № 4, с. 14–19*

## ПАРАЛЛЕЛЬНОЕ И РАСПРЕДЕЛЕННОЕ ПРОГРАММИРОВАНИЕ

*УДК 681.3*

# ПОСТРОЕНИЕ БОРТОВЫХ КОММУТИРУЕМЫХ СЕТЕЙ МИНИМАЛЬНОЙ СЛОЖНОСТИ

**© 2021 г. В. А. Костенко[a],*, А. А. Морквин[a],****

[a] *Московский государственный университет имени М.В. Ломоносова, 119991 Москва, Ленинские горы, д. 1, Россия*

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

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

Поступила в редакцию 26.10.2020 г.
После доработки 20.01.2021 г.
Принята к публикации 26.01.2021 г.

В статье сформулирована задача построения бортовой коммутируемой сети минимальной сложности необходимой для передачи периодических сообщений в реальном времени и предложены алгоритмы ее решения: построения структуры сети и системы виртуальных каналов. Приводятся результаты апробации предложенных алгоритмов для построения бортовых сетей AFDX, предназначенных для передачи исходно заданного набора периодических сообщений.

**DOI:** 10.31857/S013234742104004X

## 1. ВВЕДЕНИЕ

Современные информационно управляющие системы реального времени (ИУС РВ) являются распределенными и включают в свой состав: вычислительные ресурсы, датчики, контроллеры исполнительных устройств, устройства хранения и отображения информации, 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:

1. ARINC 651 – основные принципы построения ИУС РВ на основе ИМА [1].

2. ARINC 653 – спецификация операционных систем [2].

3. FC-AE-ASM-RT – спецификация сети информационного обмена на основе коммутируемой сети Fibre Channel [3].

4. ARINC 664 (AFDX) – спецификация сети информационного обмена на основе Ethernet [4].

Стандарт ARINC 651 регламентирует основные принципы построения ИУС РВ на основе ИМА. В соответствии с этим стандартом, единый бортовой вычислитель строится из набора стандартизованных вычислительных модулей.

Стандарт ARINC 653 регламентирует построение операционных систем, которые позволяют обеспечить выполнение программ в реальном времени и изоляцию программ различных бортовых подсистем при выполнении на едином вычислителе. Изоляция распространяется на все ресурсы, включая регистровую память, кэши центральных процессоров, шины (порты) ввода-вывода. Изоляция программ различных подсистем ИУС РВ обеспечивается введением разделов и окон. Для программ каждой бортовой подсистемы выделяется свой раздел и набор временных окон (непересекающихся интервалов времени). Расписание закрытия и открытия окон строится предварительно, до начала работы системы. Программы раздела могут выполняться только в рамках своих окон и каждому разделу выделяется необходимая память, к которой не могут обращаться программы других разделов. Программы раздела внутри окна запускаются на выполнение по мере готовности данных, в соответствии с приоритетами. Допустимо прерывание программы и ее последующее выполнение в этом окне или в одном из следующих окон раздела. Программы

различных разделов могут взаимодействовать лишь путем передачи сообщений.

Стандарт FC-AE-ASM-RT регламентирует построение сетей обмена ИУС РВ на основе Fibre Channel. Стандарт ARINC 664 (AFDX) используется при построении бортовых сетей обмена летательных аппаратов гражданской авиации. Базовая топология: коммутируемая сеть на основе стандарта Ethernet 802.3. Передача сообщений в режиме реального времени достигается введением виртуальных каналов и механизма контроля трафика по каждому каналу. Характеристики виртуальных каналов и их маршруты определяются при построении бортовой сети таким образом, чтобы гарантировано обеспечить передачу периодических сообщений по каждому виртуальному каналу в режиме реального времени, то есть сообщение должно обязательно передаваться один раз в каждом периоде.

В работе [5] 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 вычислителя 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. Это обуславливает актуальность построения бортовых сетей обмена минимально необходимой сложности.

В данной работе сформулирована задача построения бортовой коммутируемой сети минимальной сложности необходимой для передачи периодических сообщений в реальном времени и предложены алгоритмы ее решения: построения структуры сети и системы виртуальных каналов. Приводятся результаты апробации предложенных алгоритмов для построения бортовых сетей AFDX.

## 2. ЗАДАЧА ПОСТРОЕНИЯ БОРТОВОЙ СЕТИ ОБМЕНА НА ОСНОВЕ СТАНДАРТА AFDX

В работе [6] сформулирована задача построения системы виртуальных каналов для заданной бортовой сети обмена и предложен алгоритм её решения. В данной работе формулируется задача построения бортовой коммутируемой сети минимальной сложности необходимой для передачи периодических сообщений в реальном времени: построения структуры сети и системы виртуальных каналов.

Пусть даны следующие множества: $N$ – множество точек размещения оконечных систем, $K$ – множество возможных точек размещения коммутаторов, $E_{sw}$ – множество возможных дуг между коммутаторами, $E_{end}$ – множество возможных дуг между оконечными системами и коммутаторами, $V$ – множество весов дуг (длина соединения). Далее под соединением будем понимать соответствующую дугу. Для каждой дуги[1] $e \in E_{sw}E_{end}$ также 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. $T_{msg}$ (мс) – период передачи сообщения.

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

3. $J_{msg}$ (мкс) – максимальный джиттер порождения сообщения внутри периода, то есть максимальный интервал времени от начала периода, в котором может быть порождено сообщение.

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 выдачи от абонента оконечной системе-отправителю до момента получения всеми абонентами-получателями.

3. $J^*_{msg}$ (мс) – максимальный джиттер передачи сообщения, то есть разность между максимальной и минимальной длительностью передачи сообщения.

*Бортовую сеть обмена* будем описывать взвешенным графом $G^* = (N^* \cup K^*, E^*, V^*)$, где: $N^* \subseteq N$ – подмножество оконечных систем, абоненты которых либо являются отправителями, либо получателями сообщений; $K^* \subseteq K$ – = подмножество коммутаторов таких, что через каждый коммутатор проходит как минимум один маршрут передачи данных; $E^* \subseteq E_{sw} \cup E_{end}$ – дуги, по которым проходит хотя бы один маршрут передачи данных, $V^* \subseteq V$ – веса всех дуг $e \in E^*$

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

[1] Согласно стандарту [4], избыточность достигается путем использования двух независимых сетей. Для каждого виртуального канала оконечная система посылает по копии данных в обе сети. В данной работе исследуется передача данных только через одну сеть, результаты для другой сети будут аналогичными.

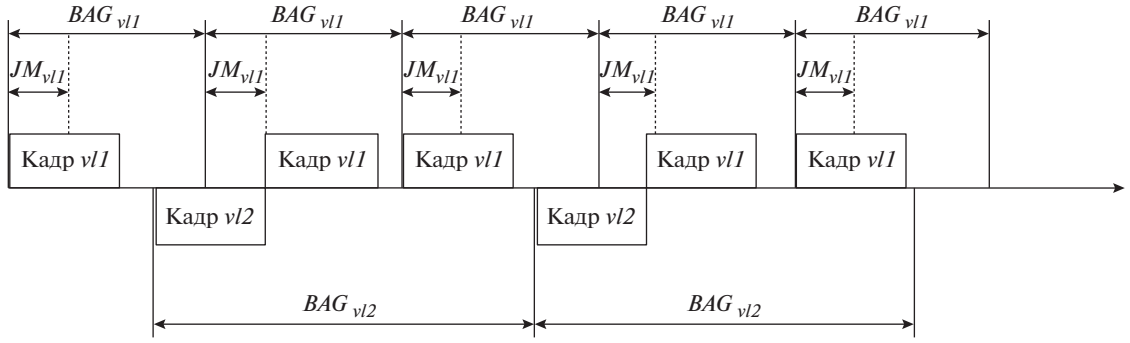


**Рис. 1.** Мультиплексирование виртуальных каналов с помощью джиттера.

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

Тогда под *максимальной сетью (полным графом)* будем понимать бортовую сеть обмена максимальной сложности, то есть бортовую сеть $G$, которая включает все коммутаторы из множества $K$ и все дуги из множества $E_{sw} \cup E_{end}$.

Построить *виртуальный канал* $vl$ значит определить для него следующие параметры:

1. $LM_{vl}$ (байт) – максимальный размер кадра, передаваемого по данному виртуальному каналу. $64 \le LM_{vl} \le 1518$ байт; размер заголовка $c = 47$ байт; расстояние между кадрами равно $gap = 12$ мкс.

2. $BAG_{vl}$ (мс) – минимальный промежуток времени между передачей кадров при нулевом джиттере порождения кадров; по стандарту это значение является степенью двойки и лежит в промежутке от 1 до 128 мс.

3. $JM_{vl}$ – максимальный джиттер порождения кадров на оконечной системе-отправителе, то есть разность между реальным временем начала выдачи кадра и временем выдачи, определенным согласно периоду. Используется при мультиплексировании виртуальных каналов.

4. $S_{vl}$ – оконечная система-отправитель кадров данного канала.

5. $D_{vl}$ – множество оконечных систем-получателей кадров данного канала.

6. $Tree_{vl}$ – маршруты передачи кадров в сети.

7. $MSG_{vl}$ – множество сообщений, передаваемых по данному виртуальному каналу и исходящих от одного абонента $S_{vl}$.

На рисунке 1 показано, что кадры виртуальных каналов $vl_1$ и $vl_2$ не могут быть переданы строго регулярно (с интервалом между кадрами, равным *BAG*). При сдвиге кадров виртуального канала $vl_1$ относительно $BAG_{vl1}$ на величину, не превышающую $JM_{vl1}$, кадры могут быть переданы, не нарушая регулярности передачи.

При построении виртуальных каналов и маршрутов для них должны выполняться следующие ограничения:

1. Суммарная пропускная способность, зарезервированная под виртуальные каналы, проходящие через дугу $e$, не превосходит его пропускной способности:

$$\forall e \in E^* : \sum_{vl \in e} \frac{LM_{vl}}{BAG_{vl}} \le R_e \qquad (1)$$

2. Частота передачи кадров каждого виртуального канала не превосходит частоты выдачи кадров в канал:

$$\forall vl \in VL : \Sigma_{msg \in MSG_{vl}} \left\lceil \frac{size_{msg}}{(LM_{vl} - c)} \right\rceil * \frac{1}{T_{msg}} \le \frac{1}{BAG_{vl}} \qquad (2)$$

Данное ограничение возникает из того, что все кадры одного сообщения *msg* должны поступить в канал до выдачи следующего сообщения *msg* то есть за период $T_{msg}$. Учитывая, что сообщение *msg* делится на количество кадров, равное $\left| \frac{size_{msg}}{LM_{vl} - c} \right|$.

3. Максимальный джиттер на оконечных системах-отправителях не должен превосходить 500 мкс[2]:

$$\forall vl \in VL : JM_{vl} \le 0.5 \text{ мкс} \qquad (3)$$

4. Максимальная длительность передачи каждого сообщения и максимальный джиттер не превосходят заданных ограничений:

$$\forall msg \in MSG : \begin{cases} Dur(msg) \le t_{msg} \\ Jit(msg) \le J^*_{msg} \end{cases} \qquad (4)$$

Здесь $Dur(msg)$ и $Jit(msg)$ – процедуры вычисления оценок длительности передачи сообщений и их джиттеров соответственно.

*Решение задачи* заключается в построении минимальной бортовой сети обмена $G^*$ и множества

[2] Данное ограничение было добавлено в задачу исходя из рекомендаций, описанных в стандарте [4].

виртуальных каналов $VL$, построенного для максимального подмножества $MSG^* \subseteq MSG$. Таким образом имеется две задачи условной оптимизации с ограничениями (1)–(4):

$$\begin{array}{cc} max(|MSG^*|) & min(S(G^*)) \\ MSG^* \subseteq MSG & G^* \in U \\ ограничения\,(1)–(4) & ограничения\,(1)–(4), \end{array}$$

где $U$ – множество всех подграфов максимальной сети. Поэтому в качестве первого критерия выберем максимизацию числа передаваемых сообщений, а вторым критерием выберем минимизацию сложности бортовой сети.

Из такого выбора критериев оптимизации следует наличие двух классов задач:

1. Такие входные данные задачи, что все сообщения можно разместить в максимальной сети.

2. Такие входные данные задачи, что все сообщения нельзя разместить даже в максимальной сети.

## 3. АЛГОРИТМ ПОСТРОЕНИЯ БОРТОВОЙ СЕТИ ОБМЕНА МИНИМАЛЬНОЙ СЛОЖНОСТИ

Алгоритм состоит из следующих шагов:

Шаг 1. Из входных данных создать максимальную сеть $G$.

Шаг 2. Для каждого сообщения из $MSG$ создать виртуальный канал с такими параметрами $LM$, $BAG$ и $JM$, чтобы выполнялось ограничение (2).

Шаг 3. Для каждого виртуального канала проверить выполнение ограничения (3). Если ограничение выполняется не для всех виртуальных каналов, то воспользоваться процедурой агрегации. Данная процедура объединяет сообщения, исходящие от одного абонента, в один виртуальный канал (при этом прежние виртуальные каналы этих сообщений удаляются) таким образом, что будут выполняться ограничения (2) и (3).

Шаг 4. Для каждого виртуального канала произвести построение маршрута с учетом ограничения (1) и суммарной длины уже используемых для передачи данных соединений, где перебор виртуальных каналов будем выполнять в порядке убывания требуемой пропускной способности:

Шаг 4.1. Выполнить процедуру построения маршрута. Если маршрут найден, перейти к шагу 5.

Шаг 4.2. Выполнить процедуру ограниченного перебора для маршрутов уже назначенных виртуальных каналов.

Шаг 4.2.1. Если маршрут построен успешно, то перейти к следующему виртуальному каналу.

Шаг 4.2.2. Если маршрут построить не удалось, назначение считается неуспешным, если в виртуальном канале передается только одно сообщение. Иначе из виртуального канала убирается сообщение с наибольшим значением $LM/BAG$ (его назначение считается неуспешным), и виртуальный канал переконфигурируется с помощью процедуры агрегации. После этого для нового виртуального канала выполняется шаг 4.1.

Шаг 5. Для каждого сообщения выполняются следующие шаги:

Шаг 5.1. Произвести вычисление длительности и джиттера передачи сообщений, затем проверить ограничение (4) и, если ограничение выполняется, то перейти к следующему сообщению.

Шаг 5.2. Произвести процедуру переконфигурации виртуального канала – попробовать перенастроить параметры $LM$, $BAG$ и $JM$ с учетом оценки полученной, на шаге 5.1. Если после выполнения процедуры, ограничение (4) выполняется, то перейти к следующему сообщению.

Шаг 5.3. Выполнить процедуру агрегации виртуальных каналов, с построением нового маршрута и проверки ограничений (1)–(4). В случае успеха прежние виртуальные каналы заменяются на новый канал. Иначе назначение сообщения считается неуспешным.

Шаг 6. Выполнить минимизацию полного графа $G$:[3]

Шаг 6.1. Для каждого соединения проверить, используется ли он в каком-либо маршруте назначенных виртуальных каналов. Если нет, то удалить это соединение из графа.

Шаг 6.2. Для каждого коммутатора проверить число соединений с другими коммутаторами. Если их число равно 0, то удалить этот коммутатор из графа.

Процедура настройки параметров виртуальных каналов [7]:

*Вход*: виртуальный канал $vl$.

*Выход*: $vl$ с заданными параметрами $LM$, $BAG$ и $JM$.

Процедура агрегации:

*Вход*: виртуальный канал $vl$.

*Выход*: агрегированный виртуальный канал для $vl$ с заданными параметрами $LM$, $BAG$ и $JM$, либо неуспех.

Процедура ограниченного перебора виртуальных каналов:

[3] На данном этапе работы алгоритма не требуется делать попытки удаления используемых в сети передачи данных соединений с последующим перестроением маршрутов виртуальных каналов, так как процедура построения маршрута учитывает не только длину соединений, но и их участие в передачи уже назначенных сообщений, то есть на каждом этапе работы процедуры сложность сети либо не изменяется, либо увеличивается на минимальную величину.

*Вход*: множество назначенных виртуальных каналов; виртуальный канал $vl$, для которого не удается построить маршрут.

*Выход*: виртуальный канал $vl$ с построенным маршрутом, либо неуспех.

Процедура вычисления максимальной длительности и джиттера передачи сообщений [8–10]:

*Вход*: сообщение *msg*, передаваемое по виртуальному каналу $vl$.

*Выход*: $Dur(msg)$, $Jit(msg)$.

Процедура переконфигурации виртуального канала:

*Вход*: виртуальный канал $vl$ с *msg*, для которого не выполняется (4) .

*Выход*: виртуальный канал $vl$ с новыми параметрами *LM*, *BAG* и *JM*, либо неуспех.

Алгоритмы выполнения этих процедур приведены в работе [6].

Процедура построения маршрута для виртуального канала:

*Вход*: граф $G$; виртуальный канал $vl$.

*Выход*: виртуальный канал $vl$ с маршрутом передачи, либо неуспех.

Шаг 1. Преобразуем граф $G$, убрав из него все соединения, остаточная пропускная способность которых меньше, чем требуемая пропускная способность.

Шаг 2. В полученном подграфе, запустить алгоритм Йена, который использует следующие 2 критерия:

• 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 виртуальных каналов. Данный критерий, используется при нахождении одного маршрута алгоритмом Дейкстры.

• Критерий для сравнения маршрутов между собой:

$$I(path) = C_1 * \sum_{path} I(e) + C_2 * cost(path, vl),$$

где $I(e)$ – критерий веса для ребра, $cost(path, vl) = \sum_{e \in path(S_{vl}, n)} \frac{LM_{vl}}{R_e} + \sum_{v \in path(S_{vl}, n), v \neq vl} \frac{LM_{vl}}{\max_{e \in path}(R_e)}$ – эвристика [6], которая оценивает стоимость пути из длительности передачи кадра текущего виртуального канала по всем каналам передачи данного пути и из оценки ожиданий передачи кадров других виртуальных каналов, $C_1, C_2 \in [0,1]$ – коэффициенты, что $C_1 + C_2 = 1$.

Среди найденных $k$ путей будем выбирать маршрут с минимальным $I(path)$.

Алгоритм остановим после нахождения кратчайшего пути до одной из оконечных систем получателей, до которых путь еще не найден.

Шаг 3. Если найдены пути до всех получателей, то вернем объединение полученных маршрутов, которое и является маршрутом виртуального канала. Иначе перейти на шаг 2.

## 4. ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ СВОЙСТВ АЛГОРИТМА

Цель экспериментального исследования – проверить эффективность предложенного алгоритма на различных классах исходных данных. Критерии эффективности: суммарная длина используемых в сети передачи данных соединений и число назначенных сообщений.

Внутри каждого класса данных, описанных в разделе 1, генерировались три типа наборов сообщений:

1. Большое количество сообщений с низкими требованиями к длительности передачи. Генерировалось 1500 сообщений размером от 1 до 1000 байт с периодом от 1 до 1000 секунд и с ограничением длительности передачи от 0.1 до 100 секунд.

2. Малое число больших сообщений. Генерировалось 100 сообщений размером от 1 кб до 1 мб с периодом от 1 до 10 секунд и с ограничением длительности передачи от 1 до 10 секунд.

3. Сообщения с высокими требованиями к длительности передачи. Генерировалось 100 сообщений размером от 1 до 1000 байт с периодом от 10 до 100 мс и с ограничением длительности передачи от 1 до 10 мс.

Запуски проводились на топологиях, используемых в реальных ИУС РВ, в которые была добавлена избыточность. Например, топология из работы [11].

Для каждой топологии генерировались все классы данных со всеми подтипами. Для каждого подтипа генерировалось 100 случайных наборов, после чего результаты усреднялись. Основные результаты исследований указаны для всех классов данных на диаграмме 1.

На основе полученных результатов экспериментов можно сделать следующие выводы:

1) Алгоритм позволяет минимизировать исходную сеть в среднем на 23%.

2) Для второго класса данных алгоритм в большинстве случаев дает максимальную сеть.

## 5. ЗАКЛЮЧЕНИЕ

Предложенный в работе алгоритм для исходно заданного множества периодических сообщений

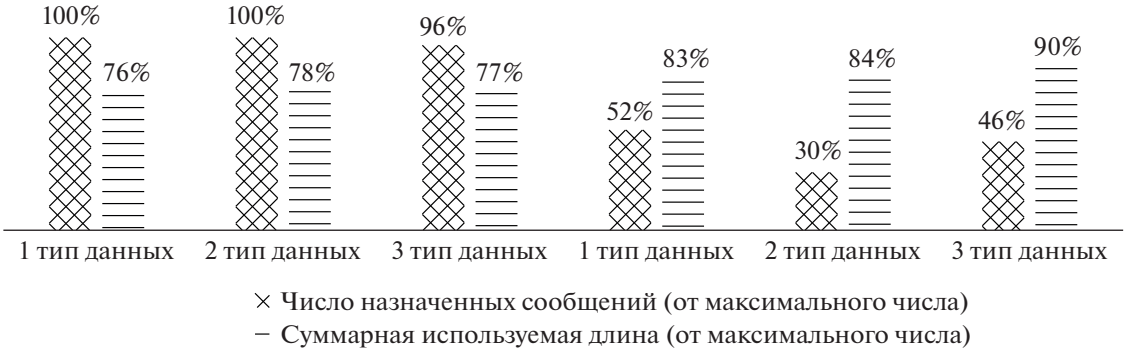


**Диаграмма 1.** Результаты для двух классов данных (слева: 1, 2, 3 типы данных для 1 класса данных; справа: 1, 2, 3 типы данных для 2 класса данных).

осуществляет построение коммутируемой сети обмена ИУС РВ минимальной сложности необходимой для передачи сообщений в реальном времени и строит множество виртуальных каналов, по которым передаются сообщения.

Алгоритм допускает настройку на особенности коммутаторов AFDX различных производителей за счет изменения алгоритмов для расчета значений параметров виртуального канала и использования различных методов для вычисления оценки максимальной длительности и джиттера передачи сообщений.

## 6. БЛАГОДАРНОСТИ

Работа выполнена при финансовой поддержке РФФИ, грант № 19-07-00614.

## СПИСОК ЛИТЕРАТУРЫ

1. ARINC 651-1 “Design Guidance for Integrated Modular Avionics”, 1997.
2. ARINC Specification 653. Airlines Electronic Engineering Committee. [PDF] (http://www.arinc.com).
3. INCITS 373. Information Technology – Fibre Channel Framing and Signaling Interface (FC-FS), International Committee for Information Technology Standards, 2003.
4. Aircraft DataNetwork. Part 7. Avionics Full Duplex Switched Ethernet (AFDX) Network // Aeronautical Radio, Inc. 2012.
5. *Костенко В.А.* Архитектура программно-аппаратных комплексов бортового оборудования // Изв. вузов. Приборостроение. 2017. Т. 60. № 3. С. 229–233.
6. *Вдовин П.М., Костенко В.А.* Организация передачи сообщений в сетях AFDX // Программирование. 2017. № 1. С. 5–20.
7. *Al Sheikh A. et al.* Optimal design of virtual links in AFDX networks. // Real-Time Systems, 2013. V. 49. № 3. P. 308–336.
8. *Scharbarg J.L., Fraboul C.* Methods and tools for the temporal analysis of avionic networks. // New trends in technologies: control, management, computational intelligence and network systems, 2010. P. 413–438.
9. *Frances F., Fraboul C., Grieu J.* Using network calculus to optimize the AFDX network // European Congress ERTS Embedded real-time software, 25–27 Jan 2006. P. 1–8.
10. *Bauer H., Scharbarg J.L., Fraboul C.* Applying Trajectory approach with static priority queuing for improving the use of available AFDX resources // Real-time systems. 2012. V. 48. № 1. P. 101–133.
11. AFDX from component to application // URL: https://www.prosoft.ru/cms/f/467416/Презентация+про+AFDX+продуктам.pdf (15.02.2020)

*ПРОГРАММИРОВАНИЕ, 2021, № 4, с. 20–30*

## ПАРАЛЛЕЛЬНОЕ И РАСПРЕДЕЛЕННОЕ ПРОГРАММИРОВАНИЕ

*УДК 004.75*

# ОСОБЕННОСТИ ВЗАИМОДЕЙСТВИЯ УСТРОЙСТВ С ИНФРАСТРУКТУРОЙ ИНТЕРНЕТА ВЕЩЕЙ НА ПРИМЕРЕ ИНФРАСТРУКТУР Amazon Web Services И Microsoft Azure

**© 2021 г. С. И. Жуков[a,*]**

[a] *Московский государственный университет имени М.В. Ломоносова*
*Научно-исследовательский вычислительный центр*
*119234, Москва, ул. Колмогорова, д. 1С4, м, Россия*

*****E-mail: serge.zhukov@auriga.com*

Поступила в редакцию 28.10.2020 г.
После доработки 18.11.2020 г.
Принята к публикации 24.11.2020 г.

Облачные инфраструктуры Amazon Web Services и Microsoft Azure поддерживают взаимодействие с IoT-устройствами (устройствами интернета вещей) по протоколу MQTT. Однако, интерфейс IoT-инфраструктуры несколько отличается, и разработка программного обеспечения для устройства, которое могло бы работать с обеими инфраструктурами, требует учета этих особенностей.

**DOI:** 10.31857/S0132347421040087

## 1. ВВЕДЕНИЕ

Контекстом данной работы явилась разработка встроенного программного обеспечения (ПО) для устройства – интеллектуального контроллера для дата-центров, реализующего электропитание, охлаждение, контроль доступа и другие функции. Конкретная функциональность контроллера зависит от набора подключенных к нему сенсоров и контроллеров более низкого уровня, а также от набора правил и логики, задаваемых пользователем. В частности, контроллер может заниматься управлением охлаждением, контролем и мониторингом доступа, сбором данных с сенсоров и их агрегированием для формирования массивов больших данных, оповещением о событиях и неисправностях.

Одной из задач при разработке встроенного ПО была реализация интерфейса “интернета вещей” (Internet of Things – IoT). При этом интеллектуальное устройство подключается через сеть Интернет к облачной инфраструктуре, поддерживающей “интернет вещей”, и взаимодействует с ней (как правило, посредством протокола MQTT). Сейчас существует несколько таких инфраструктур; наиболее популярные – Amazon Web Services (далее – AWS), Microsoft Azure (далее – Azure), Google Cloud IoT, IBM Watson IoT.

Устройство передает в облако данные, собираемые с сенсоров, для хранения и анализа, а также принимает команды управления, передаваемые через инфраструктуру. Также ПО устройства должно поддерживать механизмы для регистрации устройства в инфраструктуре и доставки IoT-конфигурации на устройство по запросу от него.

Первоначально в качестве основной облачной инфраструктуры рассматривалась AWS, однако затем возникла задача – обеспечить также возможность подключения устройства к другим инфраструктурам. При этом необходимо структурировать встроенное ПО устройства таким образом, чтобы возможно большая часть интерфейса IoT оставалась независимой от используемой инфраструктуры, а инфраструктурно-зависимый код находился бы в небольших по размеру интерфейсных модулях.

Эта задача была успешно решена для инфраструктуры Microsoft Azure, при этом выяснилось, каковы отличия IoT-интерфейса между этими инфраструктурами и как их преодолеть на уровне встроенного ПО IoT-устройства. Для доступа к инфраструктуре использовалась клиентская библиотека AWS с небольшими изменениями для поддержки Azure. Разработаны также общие механизмы для регистрации устройства в инфраструктуре и доставки IoT- конфигурации на устройство.

Сравнение возможностей обеих инфраструктур производилось в следующих направлениях:

– особенности реализации протокола MQTT;

– аутентификация устройства в IoT-инфраструктуре;

– поддержка топиков MQTT;

– сохраненное состояние устройства (Shadows и Twins);

– методы прямого действия (Azure);

– MQTT над WebSockets.

В качестве возможного подхода к решению задачи рассматривалась также архитектура Web of Things – архитектура, специально разработанная для обеспечения совместимости IoT-инфрастуктур и IoT-приложений. Однако, этот подход более применим к интерфейсу между приложением и инфраструктурой, чем к интерфейсу между инфраструктурой и устройством.

Дальнейшее изложение материала отражает анализ и сравнение возможностей инфраструктур AWS и Azure в контексте описанной выше задачи.

## 2. ОСОБЕННОСТИ РЕАЛИЗАЦИИ ПРОТОКОЛА MQTT

Протокол MQTT – протокол прикладного уровня, используемый для передачи сообщений между IoT-устройствами и облачной IoT-инфраструктурой. Для него существует промышленный стандарт [1]. Протокол основан на модели Publish-subscribe. При этом участники взаимодействия могут подписываться (subscribe) на сообщения с определенной темой. Когда кто-то из участников отправляет сообщение, оно отправляется широковещательно (publish) и его получают те участники, которые подписаны на него. Участниками взаимодействия являются как IoT-устройства, принадлежащие к определенной группе, так и исполняемые объекты внутри облачной инфраструктуры (приложения, диспетчеры событий, так называемые лямбда-функции и т.д.). Данные в протоколе MQTT передаются в текстовом структурированном формате JSON (Java Script Object Notation, [2]).

В обоих рассматриваемых облачных инфраструктурах реализован протокол MQTT версии 3.1.1 с определенными ограничениями относительно стандарта. Он реализован поверх соединения TLS между устройством и инфраструктурой, так что сообщения передаются в зашифрованном виде и поверх установленного TCP-соединения.

Основные ограничения, налагаемые обеими рассматриваемыми инфраструктурами:

– поддерживается только одно соединение между устройством и инфраструктурой;

– ограничено количество сообщений в единицу времени, которое посылает устройство;

– не поддерживаются сообщения с гарантированной однократной доставкой (определенное протоколом качество сервиса QoS=2).

## 3. АУТЕНТИФИКАЦИЯ УСТРОЙСТВА

Аутентификация устройства при установлении соединения TLS для AWS возможна только с использованием сертификата X.509 [3] (за исключением режима MQTT над WebSockets), а для Azure – по имени пользователя/паролю или также с использованием сертификата X.509. Для единообразия был выбран метод аутентификации с использованием сертификата для обеих инфраструктур.

В AWS сертификат устройства автоматически создается при регистрации устройства в IoT-инфраструктуре и хранится внутри инфраструктуры. После этого достаточно загрузить его на устройство и указать ссылку на него в конфигурационном файле клиентской библиотеки.

В Azure сертификат устройства нужно создавать вручную; при этом родительский сертификат нужно зарегистрировать в облаке. После регистрации инфраструктура принимает для аутентификации устройства любой сертификат, созданный на основе зарегистрированного родительского сертификата.

## 4. РЕГИСТРАЦИЯ УСТРОЙСТВА В ИНФРАСТРУКТУРЕ И ЗАГРУЗКА IOT-КОНФИГУРАЦИИ НА УСТРОЙСТВО

Чтобы устройство могло установить соединение с IoT-инфраструктурой, оно должно быть зарегистрировано в ней и иметь уникальный сертификат. Из соображений безопасности регистрация устройства реализуется через отдельное облачное приложение, доступное только пользователю инфраструктуры с правами локального администратора.

При регистрации устройства администратор задает имя устройства, его MAC-адрес (он уникален для каждого устройства) и тип устройства. В облачной базе данных (внутренняя база данных устройств в случае AWS, BLOB (Binary Large OBject) – контейнер в случае Azure) создается запись для данного устройства, содержащая информацию о нем, включая сертификат.

Для упрощения работы функция создания нового сертификата и конфигурационного файла для клиентской библиотеки в Azure была реализована в виде облачного приложения, реализующего HTTP REST API [4]. Регистрация нового устройства в инфраструктуре производится через это приложение. При этом оно создает соответствующие данные (сертификат, приватный ключ, конфигурационный файл), которые затем могут быть загружены на устройство путем вызова REST API. Для хранения этих данных приложение создает специальный контейнер (так называемый BLOB-контейнер) в хранилище данных Azure.

Аналогичное приложение было разработано и для инфраструктуры AWS, хотя регистрация устройства в AWS может быть сделана через AWS Console, которое представляет собой встроенный

Web-интерфейс инфраструктуры. Использование приложения позволяет осуществить регистрацию устройства “в один клик”, а также просмотреть список зарегистрированных устройств и удалить регистрацию устройства, не выходя из приложения.

Для зарегистрированного устройства доступна операция загрузки IoT-конфигурации на него. Это может быть сделано либо при инициализации устройства, либо в процессе работы устройства. Для обеих инфраструктур эта операция реализована примерно одинаково. Её общее описание можно представить в виде перечисленных далее действий.

– Встроенное программное обеспечение устройства реализует специальную функцию, которая может быть вызвана через локальный пользовательский интерфейс (Command Line Interface, touchscreen, локальный Web интерфейс). Ей передается имя устройства.

– Эта функция формирует HTTP POST запрос, включающий в его параметры имя устройства и локальный MAC-адрес. Запрос посылается на выделенный URL, принадлежащий соответствующей IoT-инфраструктуре; к этому URL привязано написанное нами облачное приложение (лямбда-функция в случае AWS, Web API приложение в случае Azure).

– Это приложение запускается инфраструктурой, получает параметры запроса и обращается к базе зарегистрированных устройств, выбирая запись по имени устройства и MAC-адресу из параметров запроса.

– В случае успешного нахождения записи, приложение формирует конфигурационный файл, включающий в себя URL оконечной точки, номера портов, сертификат и приватный ключ и другие параметры соединения, и возвращает их в ответе на запрос POST в JSON формате. Встроенное программное обеспечение обрабатывает эти данные и формирует конфигурационные файлы, чтобы подключиться к инфраструктуре после перезапуска.

– В случае неудачного поиска, приложение возвращает код ошибки 404 (страница не найдена) в ответ на запрос. Встроенное программное обеспечение сообщает пользователю об ошибке.

Часть кода, выполняемого на устройстве, для получения конфигурационного файла из облака (с использованием библиотеки CURL), приведена ниже:

```
  //
  //  Function:
  //      PopulateCloudConfigurationFilesInternal
  //  Synopsis:
  //      This function retrieves configuration file and certificates
  //      by POSTing to the designated URL in the cloud, passing it
  //      the thing name and MAC address
  //  Parameters:
  //      api_url - the designated URL which implements the API
  //      root_ca_url - the URL for the root certificate
  //      curl - the CURL context
  //      directory - where to put the configuration file and certificates
  //      thing_name - the thing name
  //      mac_address - the MAC address
  //      curl_error - the error message from CURL is placed here
  //  Return value:
  //      Error code: 0 for success, negative value for a failure
  //
static size_t write_data_function(void *buffer, size_t size, size_t nmemb,
void *userp)
{
    std::string *result = (std::string *)userp;
    result->append((const char *)buffer, size*nmemb);
    return size * nmemb;
}
static SaErrorT PopulateCloudConfigurationFilesInternal(const std::string
&api_url,
```

```
       const std::string &root_ca_url, CURL *curl, const std::string &directory,
       const std::string &thing_name, const std::string &mac_address,
    std::string &curl_error)
{
         CURLcode res;
         long httpCode = 0;

         struct curl_slist *headers = NULL;
                  headers = curl_slist_append(headers, "Content-Type: ap-
plication/json");
    headers = curl_slist_append(headers, "Accept: application/json");
       std::string post_params = "{\"thingName\":\"" + thing_name + "\",\"mac-
Address\":\"" + mac_address + "\",\"function\":\"getCertificate\"}";
    curl_easy_setopt(curl, CURLOPT_URL, api_url.c_str());
    curl_easy_setopt(curl, CURLOPT_POSTFIELDS, post_params.c_str());
    curl_easy_setopt(curl, CURLOPT_POSTFIELDSIZE, post_params.size());
    curl_easy_setopt(curl, CURLOPT_HTTPHEADER, headers);

    std::string result;
    curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, write_data_function);
    curl_easy_setopt(curl, CURLOPT_WRITEDATA, &result);
    res = curl_easy_perform(curl);
    curl_easy_getinfo(curl, CURLINFO_RESPONSE_CODE, &httpCode);
    curl_slist_free_all(headers);
    if (res || httpCode >= 300) {
        //  Download error
        if(res == 0 && httpCode == 404) {
            // Not found on POST - configuration does not exist
            return SA_ERR_HPI_NOT_PRESENT;
        }
        curl_error = res ? "Failed to obtain a certificate from the cloud: "
+ std::string(curl_easy_strerror(res)) : "HTTP Error " + std::to_string(http-
Code);
        return SA_ERR_HPI_ERROR;
    }
    try {
        json jsondata;
        std::istringstream ifile(result);
        ifile >> jsondata;
        // Subsequent code parses the JSON response in "jsondata" and places
the fields
        // of the JSON object to appropriate files
    …
        return SA_OK;
    } catch (std::exception &e) {
         dbg_print(DBG_ERROR, "Exception reading json data from \'result\':
%s\n", e.what());
        return SA_ERR_HPI_NOT_PRESENT;
    }
}
```

## 5. ИСПОЛЬЗОВАНИЕ КЛИЕНТСКОЙ БИБЛИОТЕКИ AWS НА IOT-УСТРОЙСТВЕ ДЛЯ ПОДКЛЮЧЕНИЯ К AZURE

Обе инфраструктуры предоставляют библиотеки для программного обеспечения IoT-устройств на языке C/C++, обеспечивающие доступ к функциям протокола MQTT, в том числе – установление соединения с инфраструктурой. Обе библиотеки предоставляют примерно одинаковые функциональные возможности, однако интерфейс их существенно отличается. Чтобы избежать массированной адаптации кода, вызывающего библиотечные функции, было решено использовать одну из библиотек для работы с обеими инфраструктурами.

Это оказалось возможным для клиентской библиотеки AWS [5]. Потребовались лишь небольшие изменения для заполнения полей “username” и “password” в MQTT-команде установления соединения. Эти поля не используются в AWS, но используются в Azure даже при аутентификации устройства по сертификату.

Кроме того, были добавлены распознавание и обработка жесткого разрыва соединения инфраструктурой Azure в случае ошибок протокола. В этом случае библиотека автоматически переустанавливает соединение и восстанавливает подписку на соответствующие топики MQTT, так же как и при разрыве соединения по другим причинам. Разрыв и восстановление соединения между устройством и инфраструктурой может происходить достаточно часто, в основном из-за географической удаленности устройств от соответствующей оконечной точки инфраструктуры.

## 6. ПОДДЕРЖКА ТОПИКОВ MQTT

Топик в MQTT соответствует теме публикуемого сообщения и состоит из нескольких частей разделенных символом ‘/’.

Для работы с IoT-устройствами нами была разработана система топиков, соответствующая набору функций, поддерживаемых устройствами:

– асинхронные события от устройства: посылаются на топик “<device-name>/events”;

– данные телеметрии от сенсоров устройства: посылаются на топик “<device-name>/response/sensor/<sensor-number>/reading”;

– действия типа запрос-ответ, например, запрос от устройства списка сенсоров, реализуются следующим образом: устройство подписывается на топик “<device-name>/request/#”, где “#” по соглашениям MQTT обозначает произвольное содержимое, и отвечает на топик, начинающийся с “<device-name>/response”. Например, чтобы получить общую информацию об устройстве, облачное приложение посылает MQTT-сообщение “<device-name>/request/general_information” и ожидает ответа с топиком “<device-name>/response/general_information”;

Инфраструктура AWS поддерживает использование практически любых топиков, разрешенных протоколом. Поэтому при работе с AWS набор топиков, разработанный для IoT-устройства, использовался в неизменном виде.

Azure поддерживает только топики нескольких жестко заданных форматов, при использовании устройством топика в любом другом формате инфраструктура немедленно и без объяснений разрывает MQTT-соединение.

Для MQTT-сообщений, исходящих от устройства (события, телеметрия, ответы на запросы от клиентов), единственный разрешенный формат топика в Azure имеет вид: “devices/<device-name>/messages/events/<property-bag>”. Здесь <device-name> – это имя, под которым устройство зарегистрировано в Azure IoT-инфраструктуре, а <property-bag> – набор пар “<имя>=<значение>”, разделенных символом ‘&’.

Для MQTT-сообщений, направленных устройству, формат топика имеет вид “devices/<device-name>/messages/devicebound/#”, где “#” по соглашениям MQTT обозначает произвольное содержимое.

Чтобы адаптировать принятую для IoT-устройства систему топиков для Azure, было решено применить следующую трансляцию: асинхронные события от устройств посылаются на основной топик “devices/<device-name>/messages/event”. Для топиков, используемых для запросов и ответов, а также для телеметрии, часть топика после имени устройства делается значением свойства с именем “subtopic” с заменой символов ‘/’ на ‘.’.

Например, для получения списка сенсоров на устройстве ему отправляется запрос с топиком “<device-name>/request/sensor/list”, а устройство посылает ответ с топиком “<device-name>/response/sensor/list”. В случае Azure топик запроса выглядит так: “ devices/<device-name>/messages/devicebound/subtopic=request.sensor.list”. В ответе устройства используется топик “devices/<device-name>/messages/events/subtopic=response.sensor.list”.

Данные телеметрии от сенсора номер 12 посылаются с топиком “<device-name>/response/sensor/12/reading”, который транслируется в Azure в “devices/<device-name>/messages/events/subtopic=response.sensor.12.reading”.

Выполняемая на устройстве функция трансляции топика в формат Azure приведена ниже:

```
std::string cSmrMQTTDispatcher::TranslateAzureTopic(const std::string &topic)
{
    if (startsWith(topic, "$iothub/")) {
        // Topic is already in Azure format, nothing to do
        return topic;
    }
    std::string result;
    const std::string messages_events = "/messages/events/";
    size_t pos = topic.find(messages_events);
    if (pos != std::string::npos) {
        // Topic is in standard format, need to translate
        result = topic;
        pos += messages_events.size();
        if (pos < result.size()) {
            result.insert(pos, "subtopic=");
            while((pos = result.find("/", pos)) != std::string::npos) {
                result[pos] = '.';
            }
        }
    }
    return result;
}
```

## 7. ПРОГРАММНЫЕ МЕХАНИЗМЫ SHADOWS И TWINS

Эти механизмы имеют название device shadow (“тень устройства”) в AWS и device twin (“близнец устройства”) в Azure [6, 7]. Они имеют очень похожую реализацию и предназначены для доступа к свойствам устройства в любой момент, независимо от статуса подключения устройства.

Shadow/twin представляет собой документ в формате JSON. Он включает в себя два набора свойств, а именно, desired и reported. В наборе reported хранится закешированная информация об устройстве в виде набора свойств. Когда устройство подключено к инфраструктуре, оно периодически обновляет информацию о себе в shadow/twin. Если устройство не подключено в данный момент, клиент может получить информацию от shadow/twin, сохраненную с момента последнего обновления (включая время последнего обновления).

В наборе desired хранятся запросы клиента на изменения свойств устройства (“желательные” значения свойств для клиента). В момент изменения этого набора клиентом или когда устройство подключается к инфраструктуре (если оно было отключено в момент изменения), устройство получает извещение по специальному топику, и может прочитать запрос на изменение свойств в наборе desired и изменить указанные свойства. После изменения устройство обновляет информацию в shadow/twin, перенося изменения обратно в набор reported.

Поскольку реализация shadows и twins очень похожа, функциональность встроенного программного обеспечения для работы с этими механизмами в AWS и в Azure отличается минимально. Имена свойств в Azure twins не могут содержать пробелы, в AWS shadow такое ограничение отсутствует, поэтому при адаптации к Azure пробелы в именах свойств были удалены. Для работы с shadow/twin используются специальные топики MQTT, для которых существует примерное соответствие [8, 9]. Основные топики перечислены в таблице 1.

В обеих инфраструктурах размер сохраняемых данных ограничен (8 Кб в AWS, 32 Кб в Azure), поэтому в shadow/twin сохраняются только самые основные свойства IoT-устройства. В нашем случае они включают следующее:

– идентификационные атрибуты устройства: имя, тип устройства, название модели и изготовителя, дата изготовления, аппаратная версия, серийный номер;

– сетевые адреса: IP-адрес (IPv4 или IPv6), MAC-адрес, маска подсети, адрес шлюза и DNS-серверов;

**Таблица 1.**

| Функция топика | Инициатор сообщения | AWS shadow | Azure twin |
|---|---|---|---|
| Обновление информации в секции reported | Устройство | $aws/things/<device-name>/shadow/update | $iothub/twin/PATCH/properties/ reported/?$rid=<rid> (где <rid> – уникальный числовой идентификатор запроса) |
| Обновленная информация принята | Shadow/twin | $aws/things/<device-name>/shadow/update/ accepted | $iothub/twin/res/204/?$rid= <rid> (<rid> соответствует <rid> запроса) |
| Обновленная информация отвергнута | Shadow/twin | $aws/things/<device-name>/shadow/update/ rejected | $iothub/twin/res/<HTTP-error-code>/?$rid=<rid> (<rid> соответствует <rid> запроса) |
| Обновление информации в секции desired | Shadow/twin | $aws/things/<device-name>/shadow/delta | $iothub/twin/PATCH/properties/ desired/... |
| Чтение информации из shadow/twin | Устройство | $aws/things/<device-name>/shadow/get | $iothub/twin/GET/?$rid=<rid> (где <rid> – уникальный числовой идентификатор запроса) |
| Чтение информации из shadow/twin – ответ с данными | Shadow/twin | $aws/things/<device-name>/shadow/get/ accepted | $iothub/twin/res/200/?$rid= <rid> (<rid> соответствует <rid> запроса) |
| Чтение информации из shadow/twin – запрос отвергнут | Shadow/twin | $aws/things/<device-name>/shadow/get/ rejected | $iothub/twin/res/<HTTP-error-code>/?$rid=<rid> (<rid> соответствует <rid> запроса) |

– версия встроенного программного обеспечения;

– информация о географическом местонахождении устройства.

## 8. DIRECT METHODS B AZURE

Протокол MQTT поддерживает только одностороннюю передачу сообщений и не поддерживает взаимодействие типа запрос-ответ. Однако, в Azure поверх MQTT реализован механизм direct methods (“методов прямого действия”), которые поддерживают такое взаимодействие и позволяют клиенту передать запрос на устройство и синхронно получить от него ответ (т.е. удаленно вызвать метод на устройстве) [10].

Механизм direct methods реализован следующим образом: клиент вызывает специальную API-функцию инфраструктуры, передавая ей имя устройства, имя метода, параметры в формате JSON и таймаут ожидания ответа от устройства. Инфраструктура формирует MQTT сообщение с топиком “$iothub/methods/POST/<method-name>/?$rid=<rid>” и отправляет его устройству. Здесь <method-name> – это имя метода, а <rid> – уникальный идентификатор запроса, в виде шестнадцатеричного числа.

Устройство, получив данное сообщение, обрабатывает запрос, вызывает соответствующий метод и посылает в ответ MQTT-сообщение с топиком “ $iothub/methods/res/<HTTP-result-code>/?$rid=<rid>. Здесь <rid> должен совпадать с <rid> в запросе, а <HTTP-result-code> информирует инфраструктуру о успехе или неудаче выполнения метода. Используются стандартные коды возврата HTTP. Так, в случае успеха код равен 200, если метод возвращает данные в формате JSON, или 204, если метод не возвращает данных. В случае неудачи используются коды 4xx, 5xx. Если метод возвращает данные, они передаются в составе MQTT-сообщения и возвращаются клиенту как результат выполнения метода.

Direct methods предоставляют большое удобство для синхронного взаимодействия с устройством. В случае AWS такой механизм отсутствует, и для вызова синхронных методов используется следующая парадигма:

– устройство при старте подписывается на шаблон топика “<device-name>/request/#” (где “#” по соглашениям MQTT означает произвольное содержимое);

– клиент (облачное приложение) посылает MQTT-сообщение с топиком в формате “<device-name>/request/...” (например, “<device-name>/request/sensor/list” для получения списка сенсоров или “<device-name>/request/sensor/12/get” для чтения значения сенсора);

– устройство обрабатывает запрос и посылает ответ на топик “<device-name>/response/...” (например, “<device-name>/response/sensor/list”);

– специальное правило внутри инфраструктуры перехватывает топики формата “<device-name>/response/#” и сохраняет топик и содержимое в базе данных DynamoDB с отметкой о времени сохранения;

– клиент проверяет базу данных и при появлении записи с соответствующим топиком и временем занесения использует ее как результат запроса.

Использование базы данных в рассматриваемом случае связано с тем, что для облачных приложений AWS отсутствует доступ к подписке на топики MQTT.

В случае Azure вызов синхронных методов на устройстве становится для облачных приложений намного проще и сводится к вызову соответствующего API облачной инфраструктуры. Этот API выполняется синхронно; при возврате из API приложению сразу же доступен результат вызова метода на устройстве.

Поддержка Azure direct methods на уровне устройства была реализована в общем виде и не потребовала детальной переработки каждого метода. Аналогично трансляции топиков, имя метода формируется на основе хвостовой части топика после частей “/request/” и “/response/” с заменой символа ‘/’ на ‘.’. Например, метод для получения списка сенсоров имеет имя “sensor.list”, а для получения значения сенсора с номер 12 – “sensor.12.get”.

При получении сообщения о вызове метода, имя метода транслируется в топик с добавлением “<device-name>/request/” спереди. Затем вызов метода и обычное сообщение обрабатываются общим кодом, а ответное сообщение, в случае вызова метода, транслируется в сообщение ответа на вызов метода с предварительно запомненным <rid> – уникальным идентификатором запроса.

## 9. ИСПОЛЬЗОВАНИЕ MQTT НАД WEBSOCKET

Протокол WebSocket [11] был разработан для прозрачной передачи произвольных данных поверх существующего соединения HTTP или HTTPS. Для этого используется механизм Upgrade протокола HTTP 1.1 [12], который позволяет изменить протокол для уже установленного HTTP-соединения. Для взаимодействия по протоколу WebSocket, клиент устанавливает HTTP или HTTPS соединение с сервером, и затем посылает запрос Upgrade, запрашивая “WebSocket” в качестве нового протокола. После согласия сервера на изменение протокола, обмен данными производится уже пакетами протокола WebSocket. Эти пакеты могут содержать произвольные данные. Протокол сохраняет границы сообщений при передаче (в отличие, например, от протокола TCP).

В случае MQTT над WebSocket, данными, которыми обмениваются клиент и сервер по протоколу WebSocket, являются MQTT-сообщениями. Основное преимущество использования MQTT над WebSocket заключается в том, что доступ к основному порту MQTT (8883) из внутренних сетей организаций часто бывает закрыт (из соображений безопасности), в отличие от порта HTTPS (443), доступ к которому открыт практически всегда. Поэтому использование MQTT над WebSocket уменьшает вероятность проблем при соединении устройства с инфраструктурой из-за требований IT-безопасности.

Обе инфраструктуры (AWS и Azure) поддерживают MQTT над WebSocket, и клиентская библиотека AWS поддерживает протокол WebSocket на стороне клиента. Существуют перечисленные далее отличия от обычного MQTT-соединения.

– в AWS, используется другой механизм аутентификации клиента, принятый для AWS HTTP REST API. А именнно, при активизации протокола WebSocket вместо сертификата клиент должен предоставить идентификатор доступа (access key ID) и секретный ключ (secret access key) для какого-либо пользователя, которому разрешен доступ к функциям IoT. Поэтому для поддержки WebSocket авторам пришлось дополнить механизм доставки IoT-конфигурации и сертификата на устройство. Дополнительно передаются идентификатор доступа и секретный ключ специально созданного пользователя AWS, имеющего права доступа только к функциям IoT. При использовании WebSocket, эти атрибуты передаются в клиентскую библиотеку при установлении соединения.

– в Azure, механизм аутентификации не меняется, но при активизации протокола WebSocket в заголовке Upgrade передаются несколько другие параметры по сравнению с AWS. Для успешной активизации достаточно, чтобы выполнялись следующие условия [13]:

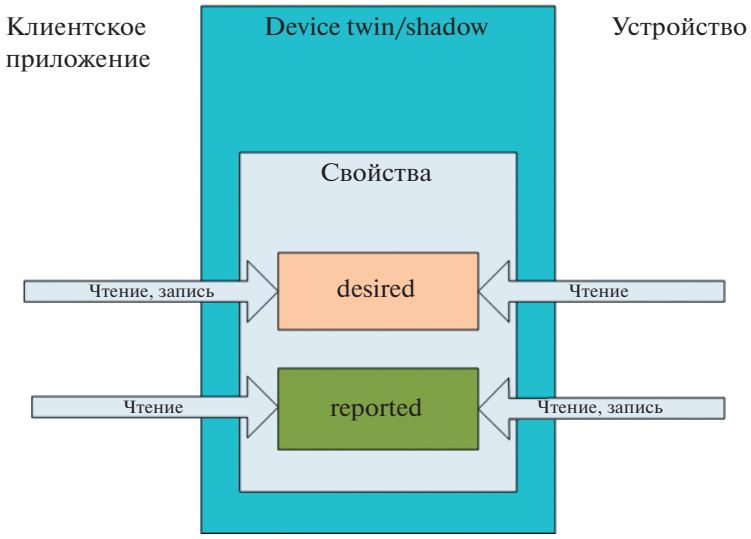


**Рис. 1.** Структура device twin/shadow.

• присутствуют обязательные параметры запроса: “Host:<endpoint address>”, “Connection: Upgrade”, “Upgrade: WebSocket”;

• параметр URL имеет значение “$iothub/websocket”;

• параметр “sec-websocket-protocol” присутствует и имеет значение “mqtt”.

Для поддержки Azure функция клиентской библиотеки, отвечающая за активизацию протокола WebSocket, была изменена: добавлен параметр “azure_mode”, и, в зависимости от значения этого параметра, заголовок Upgrade формируется с различными значениями параметров (для AWS или для Azure).

Решение о том, использовать ли стандартное соединение MQTT или MQTT над WebSockets, принимает пользователь при загрузке IoT-конфигурации на устройство. Соответствующий флаг записывается в IoT-конфигурацию. По умолчанию используется стандартное соединение; но если доступ к стандартному порту MQTT запрещен сетевым файрволлом, то можно использовать MQTT над WebSockets.

## 10. СРАВНЕНИЕ С АРХИТЕКТУРОЙ WEB OF THINGS

Архитектура Web of Things (WoT, [14]) разработана консорциумом W3C для обеспечения совместимости между различными IoT-платформами и IoT-приложениями. Это абстрактная архитектура, компоненты которой определяются следующими дочерними спецификациями:

– Спецификация Web of Things (WoT) Thing Description. [15] Эта спецификация задает стандартную модель данных для описания интерфейса IoT-устройства, в терминах свойств (properties), действий (actions) и событий (events), и в формате JSON. Приложение, работающее с устройством, может читать и записывать свойства, вызывать действия и подписываться на события (и получать их). Описание хранится вместе с устройством или отдельно от него, но доступно для считывания приложениями.

– Спецификация Web of Things (WoT) Binding Templates [16]. Этот документ описывает привязку модели данных и действий над ней к конкретным протоколам. Существуют привязки к протоколам HTTP, CoAP и MQTT. Например, в случае протокола HTTP, для каждого свойства задается URL, и чтение этого свойства отображается на операцию GET по этому URL, а запись – на операцию PUT по этому URL. Вызов действия отображается на операцию POST по соответствующему URL с определенными параметрами.

– Спецификация Web of Things (WoT) Scripting API [17]. Эта спецификация позволяет задавать на устройствах программную логику на языке Java Script, подобно тому как это делается в Web-браузерах. Это необязательный компонент архитектуры, потому что не все устройства имеют возможность содержать в себе интерпретатор JavaScript.

– Спецификация Web of Things (WoT) Security and Privacy Guidelines [18]. Этот документ содержит указания по защищенной реализации и конфигурации устройств, и рассматривает вопросы безопасности, которые требуют внимания при реализации WoT-систем.

Архитектура WoT позволяет разрабатывать облачные приложения, которые не знают заранее интерфейс устройств, с которыми они взаимо-

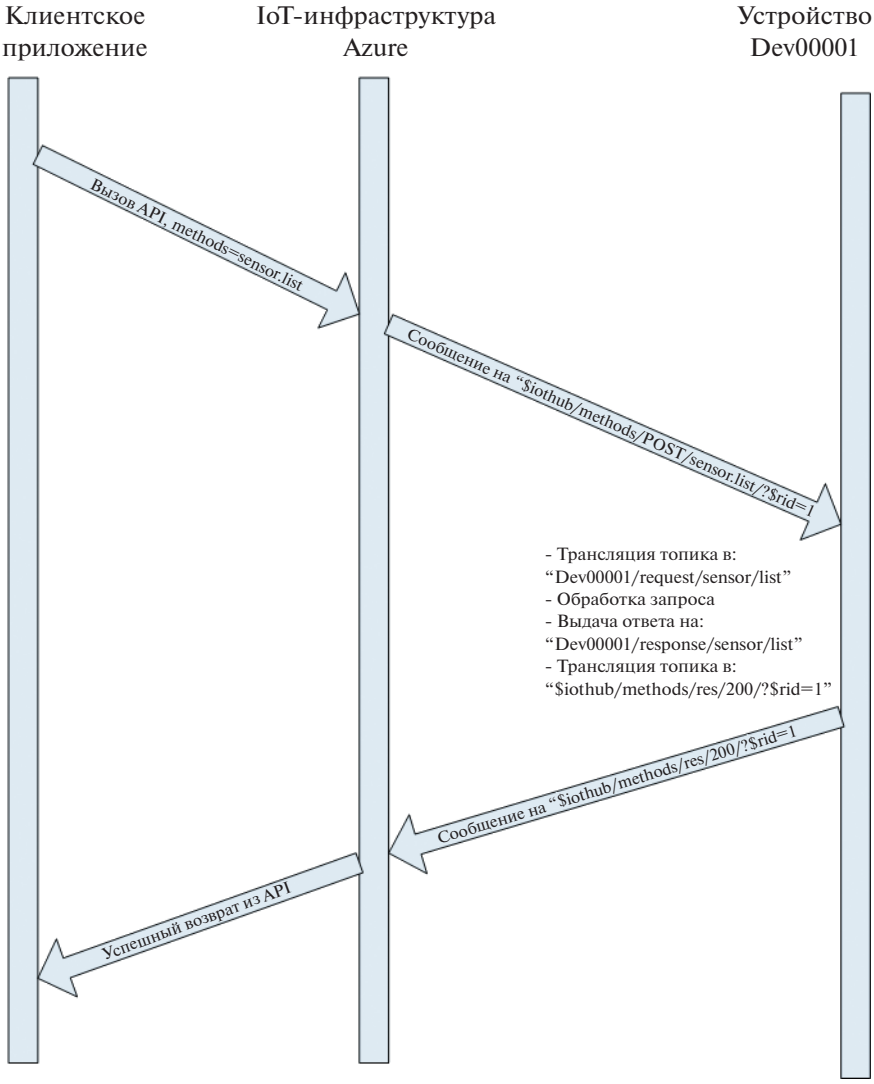


**Рис. 2.** Диаграмма взаимодействия при вызове direct method.

действуют – они получают эту информацию из описания интерфейса устройства. В этом смысле архитектура улучшает совместимость между приложениями и устройствами. Но, с точки зрения переносимости устройства, использующего протокол MQTT, между инфраструктурами, архитектура WoT не дает особых преимуществ, из-за следующих обстоятельств:

– Архитектура WoT основана на принципах REST API, и не очень хорошо приспособлена к модели Publish/Subscribe, на которой основан протокол MQTT; например, синхронное чтение свойств не поддерживается в этой модели. Как правило, в примерах описаний WoT-устройств с отображением на MQTT используются события и действия, но не используются свойства;

– В отображении на протокол MQTT топики статически отображаются на компоненты URL в описании. Учитывая, что в Azure множество поддерживаемых топиков сильно отличается от AWS, сделать единое описание устройства для обеих инфраструктур не получится;

– Расширения MQTT для конкретных инфраструктур (например direct methods в Azure) не поддерживаются архитектурой WoT.

Таким образом, технические решения, описанные в данной статье, актуальны и с учетом существования архитектуры WoT.

Поддержка архитектуры WoT, в частности, создание описания устройства для рассматриваемого контроллера имеет смысл как перспективный проект, в совокупности с реализацией взаимодействия с IoT-инфраструктурой на основе протокола HTTP/HTTPS, в дополнение к MQTT.

## 11. ЗАКЛЮЧЕНИЕ

Итак, выше рассмотрены некоторые отличия между IoT-инфраструктурами AWS и Azure, с точки зрения IoT-устройства, которое взаимодействует с инфраструктурой по протоколу MQTT. Можно сказать, что эти отличия довольно существенны, и что в каких-то аспектах имеет преимущество AWS (гибкость системы топиков), а в каких-то – Azure (direct methods). Тем не менее, при разработке программного обеспечения IoT-устройства, поддерживающего обе инфраструктуры, удалось сохранить большую часть кода независимой от инфраструктуры и локализовать зависимости в нескольких небольших интерфейсных модулях.

Разработанные технические решения можно будет использовать в будущем для подключения устройства к другим инфраструктурам, например, Google Cloud IoT. Также в перспективе – включение устройства в архитектуру Web of Things с созданием описателя устройства в соответствии с спецификацией WoT Thing Description.

## СПИСОК ЛИТЕРАТУРЫ


1. MQTT Version 3.1.1. OASIS Standard. http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html
2. ECMAScript® 2020 Language Specification. https://www.ecma-international.org/publications/files/ECMA-ST/ECMA-262.pdf
3. Recommendation ITU-T X.509. Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks. https://www.itu.int/rec/T-REC-X.509
4. Alex Rodriguez. RESTful Web Services. https://developer.ibm.com/articles/ws-restful/
5. AWS IoT C++ Device SDK. https://github.com/aws/aws-iot-device-sdk-cpp/tree/release
6. Device Shadow Service for AWS IOT. https://docs.aws.amazon.com/iot/latest/developerguide/iot-device-shadows.html
7. What is Azure Digital Twins? https://docs.microsoft.com/en-us/azure/digital-twins/overview
8. Understand and use device twins in IoT Hub https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide-device-twins
9. Shadow MQTT Topics. https://docs.aws.amazon.com/iot/latest/developerguide/device-shadow-mqtt.html
10. Understand and invoke direct methods from IoT Hub https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide-direct-methods
11. RFC 6455. The WebSocket Protocol. https://tools.ietf.org/html/rfc6455
12. Protocol Upgrade Mechanism. https://developer.mozilla.org/en-US/docs/Web/HTTP/Protocol_upgrade_mechanism
13. MQTT-over-WebSockets needs to explain how to set up the WS connection https://github.com/MicrosoftDocs/azure-docs/issues/21306
14. Web of Things (WoT) Architecture. W3C Recommendation 9 April 2020. https://www.w3.org/TR/2020/REC-wot-architecture-20200409/
15. Web of Things (WoT) Thing Description. W3C Recommendation 9 April 2020. https://www.w3.org/TR/2020/REC-wot-thing-description-20200409/
16. Web of Things (WoT) Binding Templates. W3C Working Group Note 30 January 2020. https://www.w3.org/TR/2020/NOTE-wot-binding-templates-20200130/
17. Web of Things (WoT) Scripting API. W3C Working Draft 28 October 2019. https://www.w3.org/TR/2019/WD-wot-scripting-api-20191028/
18. Web of Things (WoT) Security and Privacy Guidelines. W3C Working Group Note 6 November 2019. https://www.w3.org/TR/2019/NOTE-wot-security-20191106/

*ПРОГРАММИРОВАНИЕ, 2021, № 4, с. 31–44*

## ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

*УДК 004.056.53*

# МОДЕЛЬ ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ, СФОРМИРОВАННЫХ АЛГОРИТМАМИ ШИФРОВАНИЯ И СЖАТИЯ ДАННЫХ

**© 2021 г. А. В. Козачок[a],*, А. А. Спирин[a],****

[a] *Академия ФСО, 302034 Орел, ул. Приборостроительная, 35, Россия*

**E-mail: a.kozachok@academ.msk.rsnet.ru*

***E-mail: spirin_aa@bk.ru*

Поступила в редакцию 03.03.2021 г.
После доработки 16.03.2021 г.
Принята к публикации 17.03.2021 г.

Задача классификации источников данных, обладающих высокой энтропией, в области информационной безопасности занимает одну из ключевых позиций. В настоящее время существуют способы классификации зашифрованных и сжатых последовательностей, которые в основном используют цифровые сигнатуры или служебную информацию в случае ее передачи. В работе проведен анализ исследований в области классификации зашифрованных и сжатых данных и разработана модель зашифрованных и сжатых последовательностей. Практические эксперименты свидетельствуют о высокой точности предложенного подхода и позволяют сделать вывод об улучшении существующих методов классификации зашифрованных и сжатых данных. Предложенный способ может быть внедрен в системы защиты данных от утечек либо в корпоративные системы электронной почты для анализа отправляемых за контролируемый периметр организации вложений.

**DOI:** 10.31857/S0132347421040051

**Цель исследования** – разработать модель псевдослучайных последовательностей, сформированных алгоритмами шифрования и сжатия данных, позволяющую наиболее точно отразить статистические свойства указанных последовательностей.

**Метод исследования** – статистический анализ данных, математическая статистика, машинное обучение.

**Результат исследования** – проведен анализ исследований, направленных на решение задачи классификации зашифрованных и сжатых последовательностей в области информационной безопасности. Разработана модель псевдослучайных последовательностей, сформированных алгоритмами шифрования и сжатия данных, учитывающая их статистические признаки: распределение байт и частоты встречаемости подпоследовательностей ограниченной длины, представляющие собой новое вероятностное пространство. Приведено обоснование выбора статистических признаков, использующихся в модели псевдослучайных последовательностей. Проведены эксперименты по определению гиперпараметров классификатора на сформированном наборе данных из зашифрованных и сжатых файлов без учета их заголовков. Определены ограничения, используемые в модели псевдослучайных последовательностей, заключающиеся в условии равенства длины анализируемых псевдослучайных последовательностей около 600 кбайт. Проведены эксперименты по определению влияния статистических признаков, участвующих в формировании модели псевдослучайных последовательностей, на точность классификации. Полученные практические результаты позволяют классифицировать зашифрованные и сжатые данные с точностью 0.97.

## 1. ВВЕДЕНИЕ

Задача классификации зашифрованных и сжатых данных занимает в информационной безопасности одну из ключевых позиций: обнаружение зашифрованного вредоносного ПО, расследования в области компьютерной криминалистики, анализ трафика и защита данных от утечек.

Современные методы обнаружения зашифрованных данных опираются на энтропийный подход, который демонстрирует низкую точность классификации высокоэнтропийных данных: зашифрованной и сжатой информации, изображений, выходных последовательностей кодировщиков.

В последнее время особенно остро стоит задача предотвращения утечки конфиденциальных данных из корпоративной сети. Отчет экспертно-

аналитического центра группы компаний InfoWatch [1] свидетельствует о высокой доле (более 79%) внутренних нарушителей как источников инцидентов информационной безопасности в России.

В работе [2] отмечается, что угрозы, вызванные внутренним нарушителем, являются самым опасными и наиболее распространенными для широко круга организаций, в том числе для государственных учреждений. Вредоносные действия в данном случае осуществляются доверенными лицами внутри организаций, что приводит к существенному ущербу.

Решение задачи обнаружения зашифрованных данных является критически важным в различных областях обеспечения информационной безопасности: противодействие утечке конфиденциальных данных и обнаружение вредоносного ПО, детектирование DDoS-атак, решение задач компьютерной криминалистики, расследование инцидентов информационной безопасности. В работе [3] осуществляется обзор методов глубокого анализа данных и обнаружения зашифрованного трафика. Авторы делают вывод о неспособности методов глубокого анализа пакетов производить классификацию зашифрованного трафика, поскольку шифрование изменяет содержание информации, однако, по мнению авторов, оно не изменяет сетевые характеристики потоков передачи данных. Кроме того, отмечается необходимость применения статистических подходов, не использующих содержание передаваемых данных.

Верно классифицируя выходные последовательности алгоритмов шифрования и сжатия данных возможно улучшить существующие методы обнаружения утечки защищаемой информации.

Статья организована следующим образом: в первой части рассмотрены существующие подходы к классификации зашифрованного трафика, обнаружению DDoS-атак, искаженных файлов и файлов вредоносного ПО, сформулирована решаемая задача. Во второй части представлена разработанная модель зашифрованных и сжатых последовательностей на основе частот встречаемости подпоследовательностей длины 9 бит и распределения байт. Приведены результаты практических экспериментов.

## 2. ПОДХОДЫ К КЛАССИФИКАЦИИ ЗАШИФРОВАННЫХ И СЖАТЫХ ДАННЫХ

Исследования по классификации данных в предметной области информационной безопасности условно возможно разделить на 3 группы: подходы к классификации зашифрованного, сжатого и открытого трафиков данных информационно-телекоммуникационных сетей; обнаружение DDoS-атак; определение типа передаваемых данных, вредоносного программного обеспечения. Обобщенный анализ рассмотренных работ представлен в таблице 1.

### *2.1. Задача классификации зашифрованных и сжатых данных*

Существующие DLP-системы выполняют анализ служебной информации, присущей передаче данных, либо на основе поиска различных сигнатур и регулярных выражений непосредственно в данных. В ряде работ отмечается невозможность обнаружения конфиденциальных данных в зашифрованном или сжатом виде [31, 32].

Ряд исследователей отмечают, что не существует эффективных и точных методов классификации источников с высокой энтропией, например алгоритмов шифрования и сжатия данных. [33, 34].

Внутреннему нарушителю доступны средства шифрования и сжатия, что позволяет сделать вывод об актуальности задачи классификации зашифрованных и сжатых данных. Рассмотренные подходы не могут являться надежным решением для обнаружения передачи зашифрованных данных по ряду причин:

1. Методы анализа трафика не применимы, поскольку нельзя допустить передачу данных за контролируемый периметр организации. Необходимо разработать средства анализа данных до их отправки вовне, например после их загрузки на сервер электронной почты.

2. Подходы с применением нейронных сетей в основном требуют значительно большего времени на обучение классификатора, чем другие алгоритмы машинного обучения. Как правило, в данных подходах применяются заголовки файлов, содержащие “магические” байты, обладающие высокой дискриминирующей способностью.

3. Рассмотренные энтропийные и иные подходы также оперируют заголовками файлов, содержащими “магические” байты.

Таким образом, можно сформулировать требования, предъявляемые к методу классификации зашифрованных и сжатых данных:

1. Применение алгоритмов машинного обучения с минимальным временем обучения классификатора.

2. Использование статистических подходов, не учитывающих заголовки файлов и специальные дискриминирующие сигнатуры.

3. Возможность применения на серверной стороне, до момента передачи за контролируемый периметр организации.

4. Минимальное время выполнения классификации последовательности.

**Таблица 1.** Обобщенный анализ рассмотренных работ

| Источник | Год | Объект | Признаки | Мат. аппарат | Точность |
|---|---|---|---|---|---|
| **Задача классификации трафика** | | | | | |
| [4] | 2016 | Трафик | Энтропия | Дерево решений | 0.981 |
| [5] | 2017 | | Служебная информация | Цепи Маркова 2-го порядка | 0.9122 |
| [6] | 2017 | | Служебная информация, распределение байт | kNN | 0.952 |
| [7] | 2018 | | Служебная информация, распределение байт | Случайные поля Маркова, СНС | 0.979 |
| [8] | 2019 | | Служебная информация, трассы DNS, характеристика обмена сертификатами | XGBoost | 0.987 |
| [9] | 2019 | | Служебная информация и статистический анализ | СНС, ГСС, автокодировщики | Обзор методов |
| [10] | 2019 | Файлы, трафик | Энтропия | МОВ, СЛ | Файлы – 0.72 Трафик – 0.979 |
| [11] | 2020 | Трафик | Служебная информация | СНС, СЛ, ДР, XGBoost, | 0.96 |
| [12] | 2020 | | Служебная информация, распределение байт | Автокорреляция | 100% |
| [13] | 2020 | | Служебная информация | Скрытые цепи Маркова, модель смеси Гауссовых распределений | Трафик сети Tor – 0.999 |
| [14] | 2020 | | Служебная информация, статистические хар-ки | МОВ | 0.8 |
| [15] | 2020 | | Служебная информация, статистические хар-ки | СЛ | 0.882 |
| **Задача обнаружения DDoS-атак** | | | | | |
| [16] | 2017 | DDoS | Служебная информация | Кластерный анализ | 0.998 |
| [17] | 2017 | | Служебная информация | Кластерный анализ | 0.989 |
| [18, 19] | 2018–2019 | | Служебная информация | Теория вероятностей | 0.97 |
| [20] | 2020 | | Служебная информация, статистические признаки | СЛ, ДР | 0.98 |
| [21] | 2018 | | Служебная информация | kNN | 0.992 |
| **Задача классификации файлов** | | | | | |
| [22] | 2017 | Файлы | Статистические признаки | SVM | 0.607 |
| [23, 24] | 2017 | | Распределение байт | ГА, НС | 98.21–100% |
| [25] | 2018 | ВПО | Цепочки команд (n–граммы) | Скрытые цепи Маркова | |
| [26] | 2019 | Файлы | Распределение байт | СНС (VGG-16, ResNet) | 0.999 |
| [27] | 2019 | | Частота встречаемости слов | XGBoost | 98.84% |
| [28] | 2019 | Файлы, трафик | Тесты NIST, энтропия блоков данных | HEDGE (High Entropy DistinGuishEr) | 0.72 |
| [29] | 2020 | Файлы | Распределение байт | НС | 80–100% |
| [30] | 2020 | | Энтропия | Скрытые цепи Маркова | 52% |

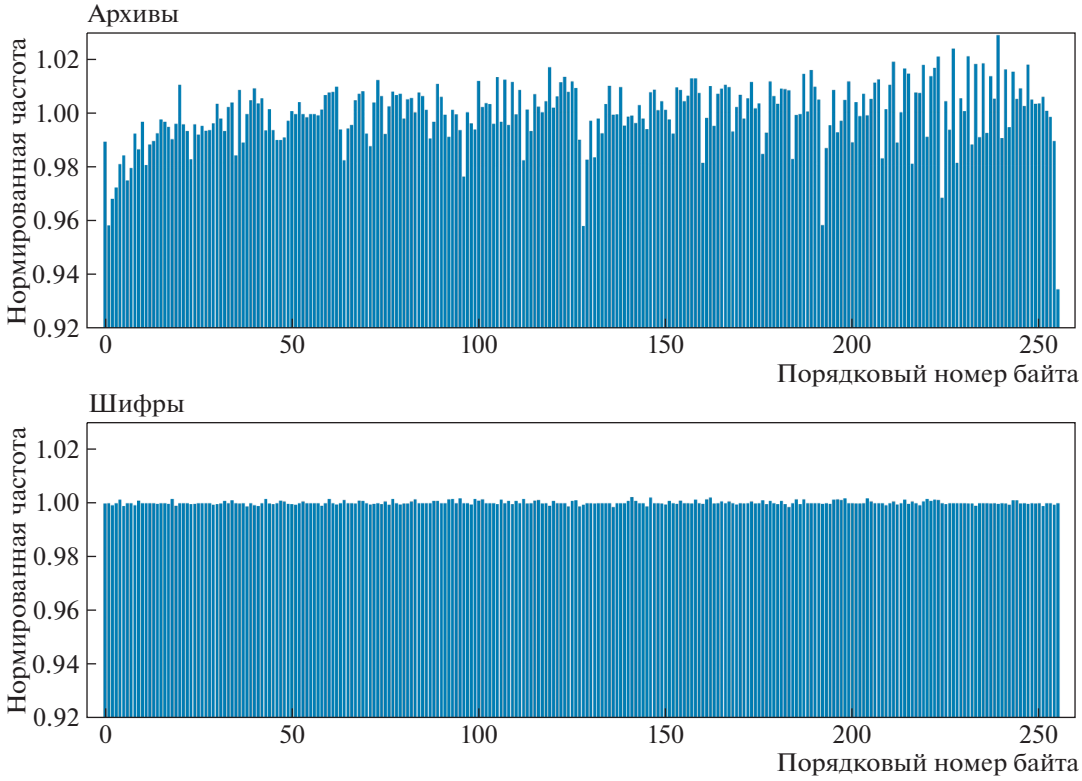


**Рис. 1.** Распределение байт для зашифрованных и сжатых последовательностей.

В формальном виде задача бинарной классификации псевдослучайных последовательностей (ПСП – последовательностей, обладающих равномерным распределением байт) может быть представлена в выражении (1):

$$F(V(s)) = k, \quad k \in \{0,1\}, \tag{1}$$

где $V(s)$ – модель ПСП, $s$ – анализируемая ПСП, $k$ – класс ПСП, $F$ – функция классификации ПСП.

## 3. МОДЕЛЬ ПСП

Для построения модели ПСП было сформировано 2000 файлов, содержащих осмысленный текст. Далее каждый из них был преобразован алгоритмами шифрования (OpenSSL[1]: AES, 3DES, Camellia, RC4 и ГОСТ 34.12 “Кузнечик” в режиме простой замены) и сжатия данных (WinRAR[2]: RAR, ZIP и 7Zip[3]: 7Z, XZ, GZ, BZ2). Размер файлов после обработки алгоритмами преобразования данных составил 600 кбайт.

Учитывая, что структура файлов имеет заголовочную часть, которую возможно модифицировать для маскирования информации, содержащейся в ней, данный способ может быть использован злоумышленниками для передачи конфиденциальной информации. В работе [35] авторы сделали вывод о необходимости удаления заголовков файлов, так как они содержат цифровые сигнатуры, позволяющие классифицировать тип данных с высокой точностью. При построении модели ПСП были отброшены первые 10 кбайт файлов, чтобы исключить влияние заголовков файлов и “магических” байт на процедуру классификации.

На первом этапе построения модели ПСП были произведены оценки распределения байт последовательностей двух классов, нормированных по среднему значению частоты встречаемости байт в выборке данных, результаты представлены на рисунке 1.

Визуально, исходя из рисунка 1, возможно сделать вывод о том, что распределение байт зашифрованных ПСП более равномерно, чем распределение байт сжатых ПСП.

Для оценки полученных распределений частот встречаемости байт в ПСП была проведена оценка их соответствия равномерному распределению согласно критерию Хи-квадрат, определяемому выражением (2):

[1] https://www.openssl.org/ (дата обращения 08.02.2021)
[2] https://www.win-rar.com/ (дата обращения 08.02.2021)
[3] https://www.7-zip.org/ (дата обращения 08.02.2021)

**Таблица 2.** Проверка гипотезы о равномерности распределения байт в ПСП

| № п/п | ПСП | Референсное распределение | Критическое значение критерия | Расчетное значение критерия | P-value |
|---|---|---|---|---|---|
| 1 | Архивы | Равномерное | 293.248 | 50.811 | 1 |
| 2 | Шифры | Равномерное | 293.248 | 0.526 | 1 |

$$\begin{cases} O_{rc} = \sum\limits_{i=1}^{k} \dfrac{(x_i - m_i)^2}{m_i} \\ X^2 = \sum\limits_{r=1}^{R}\sum\limits_{c=1}^{C} \dfrac{(O_{rc} - E_{rc})^2}{E_{rc}} \end{cases}, \quad (2)$$

где $m_i$ – математическое ожидание частоты появления байта $i \in \{0,\ldots,255\}$ в анализируемой ПСП, для равномерного закона распределения и длины ПСП = 600 кбайт $m_i = \text{const} = 2400$; $x_i$ – значение частоты появления байта $i \in \{0,\ldots,255\}$ в анализируемой ПСП; $O_{rc}, E_{rc}$ – наблюдаемое и ожидаемое значения критерия на выборке данных размером $r$ строк и $c$ столбцов. Результаты представлены в таблице 2.

Полученные значения позволяют сделать вывод о равномерном характере распределений байт в рассматриваемых последовательностях и считать их псевдослучайными. Признаковое пространство на основе распределения байт может быть представлено выражением (3):

$$V_{Bytes} = \langle F(b_i)\rangle, \quad i \in \langle 0,\ldots,255\rangle, \quad (3)$$

где $F(b_i)$ – значение частоты появления байта $i$ в анализируемой ПСП.

Результаты экспериментов по определению точности классификации зашифрованных и сжатых последовательностей различными алгоритмами машинного обучения при использовании признаков, являющихся значениями частот распределение байт представлены в выражении (4).

$$\begin{cases} Acc_{RF}(V_{Bytes}) = F(V_{Bytes}) = 0.9 \\ Acc_{DT}(V_{Bytes}) = F(V_{Bytes}) = 0.88 \\ Acc_{kNN}(V_{Bytes}) = F(V_{Bytes}) = 0.89 \end{cases}, \quad (4)$$

где $V_{Bytes}$ – признаковое пространство, определяемое выражением (3), $Acc_{RF,DT,kNN}$ – метрики "точность классификации" ПСП соответствующим алгоритмом машинного обучения.

Полученные результаты позволяют построить классификатор зашифрованных и сжатых последовательностей, однако точность классификации недостаточно высока. Распределение байт для данных, обладающих высокой энтропией, например для зашифрованных и сжатых данных, подчиняется равномерному закону распределения. В литературе встречаются исследования, приводящие к выводу о невозможности классификации зашифрованных и сжатых данных с высокой точностью при использовании распределения байт [36], что объясняется другой предметной областью исследования, связанной с обнаружением вредоносного ПО, где размеры анализируемых данных недостаточно велики.

Кроме того, для построения модели ПСП недостаточно лишь распределения байт, так как они стремятся к равномерному распределению. Необходимо обнаружить новое вероятностное пространство признаков, позволяющее создать наиболее точную модель ПСП, вследствие чего были проведены расчёты средних значений энтропии Шеннона для зашифрованных и сжатых последовательностей согласно выражению (5):

$$H = -\sum_{i=0}^{255} p_i * \log_2 p_i, \quad (5)$$

где $p_i$ – вероятность появления байта $i \in \{0,\ldots,255\}$ в анализируемой последовательности.

Были получены средние значения энтропии и значения некоторых статистических параметров распределений энтропии байт в анализируемых ПСП, результаты представлены в таблице 3, полученные согласно выражению 6:

$$\begin{cases} E_{mean} = \dfrac{\Sigma E(b_i)}{256} \\ E_{sko} = \sqrt{\dfrac{\Sigma(E(b_i) - E_{mean})^2}{256}} \\ E_{median} = X_{median} + i_{median} * \dfrac{\dfrac{\Sigma E(b_i)}{2} - Sum_{median-1}}{E(b_i)} \end{cases}, \quad (6)$$

где $E(b_i)$ – значение энтропии байта $i$ в анализируемой ПСП; $i_{median}$ – размер медианного интервала; $X_{median}$ – нижняя граница медианного интервала, $Sum_{median\ -\ 1}$ – сумма значений энтропии байт, предшествующих медианному интервалу.

Проведены эксперименты по поиску более дискриминирующих статистических признаков, чем распределение и значения энтропии байт. В некоторых работах [28] применяются статистические те-

**Таблица 3.** Статистические признаки распределения значений энтропии байт в ПСП

| № п/п | Признак | Архивы | Шифры |
|---|---|---|---|
| 1 | $E_{mean}$ | 5.54424 | 5.54496 |
| 2 | $E_{sko}$ | 0.00106 | 0.00002 |
| 3 | $E_{median}$ | 5.54468 | 5.54496 |

**Таблица 4.** Оценка точности классификации алгоритмами машинного обучения при использовании модели ПСП на основе тестов NIST

| № п/п | Алгоритм | Accuracy |
|---|---|---|
| 1 | Random Forest | 0.643 |
| 2 | Decision Tree | 0.575 |
| 3 | kNN | 0.684 |

сты NIST[4] для извлечения признаков из анализируемых данных.

Далее для построения модели ПСП были использованы статистические тесты NIST. Элементами модели являлись значения p-value, полученные в результате прохождения статистических тестов, диаграмма размаха полученных средних значений p-value представлена на рисунке 2.

В результате было установлено, при уровне значимости $\alpha = 0.05 \times 100\%$ зашифрованных последовательностей прошли все тесты на случайность, а сжатые последовательности – 98%. На основании выявленных закономерностей (схожие значения энтропии, прохождение тестов NIST на случайность) было принято решение обозначить зашифрованные и сжатые последовательности как псевдослучайные.

Таким образом, модель ПСП на основе распределений значений p-value тестов NIST представляет собой усредненные значения p-value, полученные в результате выполнения 11 статистических тестов: частотный побитовый ($p_{freq}$), блочный частотный ($p_{Bfreq}$), тесты кумулятивных сумм ($p_{CS1}, p_{CS2}$, тест на последовательность одинаковых бит ($p_{Runs}$), тест на самую длинную последовательность единиц в блоке ($p_{LRuns}$), тест рангов бинарных матриц ($p_{Rank}$), спектральный тест ($p_{FFT}$), тест на обнаружение неперекрывающихся шаблонов ($p_{NOT}$), тест на обнаружение перекрывающихся шаблонов ($p_{OT}$), тест приблизительной энтропии шаблонов ($p_{AEnt}$). Таким образом, признаковое пространство на основе тестов NIST является вектором, определяемым выражением (7):

$$V_{NIST} = \langle p_{freq}, p_{Bfreq}, p_{CS1}, p_{CS2}, p_{Runs}, p_{LRuns}, p_{Rank}, p_{FFT}, p_{NOT}, p_{OT}, p_{AEnt} \rangle, \quad (7)$$

где $p_i$ – средние значения p-value по выборке данных для соответствующих тестов NIST.

Для оценки признакового пространства были проведены эксперименты, результаты представлены в таблице 4.

При использовании модели на основе тестов NIST точность классификации ПСП по сравнению с моделью на основе распределения байт значительно ухудшилась. Данный факт объясняется тем, что тесты NIST направлены на обнаружение закономерностей в анализируемых последовательностях, проверку их на случайность возникновения символов. Кроме того, процедура тестирования предполагает получение результатов в 10 возможных интервалах и вычислении p-value на основе табличных значений (критерий Хи-квадрат и равномерное распределение), что стирает статистические различия между рассматриваемыми ПСП, но позволяет оценить гипотезу о случайности анализируемых данных.

В статистических тестах NIST присутствует тест на проверку непересекающихся шаблонов. Методика тестирования предполагает разбиение анализируемой последовательности на $N$ блоков данных длиной $M$ бит и поиска в них бинарных шаблонов длиной $m$ бит (подпоследовательностей). В случае обнаружения подпоследовательности его частота увеличивается на 1, и окно поиска сдвигается на следующий бит после последнего бита найденного шаблона. Для подтверждения гипотезы о случайности последовательности вычисляются математическое ожидание и дисперсия частоты встречаемости шаблонов в теоретически случайном распределении согласно выражениям (8) и (9):

$$\mu = \frac{M - m + 1}{2^m}, \quad (8)$$

$$\sigma^2 = M * \left(\frac{1}{2^m} - \frac{2m-1}{2^{2m}}\right)\frac{M - m + 1}{2^m}, \quad (9)$$

где $\mu$ – математическое ожидание частоты встречаемости шаблона длиной $m$ бит в теоретически случайной последовательности, разделенной на $M$ блоков даных.

Далее вычисляется значение критерия Хи-квадрат по полученным значениям частот встречаемости шаблонов подпоследовательностей согласно выражению (2).

Итоговое решение о прохождении теста принимается на основе вычисленного значения p-value согласно неполной гаммы-функции, выражение (10):

[4] A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, National Institute of Standards and Technology (NIST) Special Publication 800-22, Rev. 1a, Apr. 2010. https://csrc.nist.gov/publications/detail/sp/800-22/rev-1a/final (дата обращения 08.02.2021)

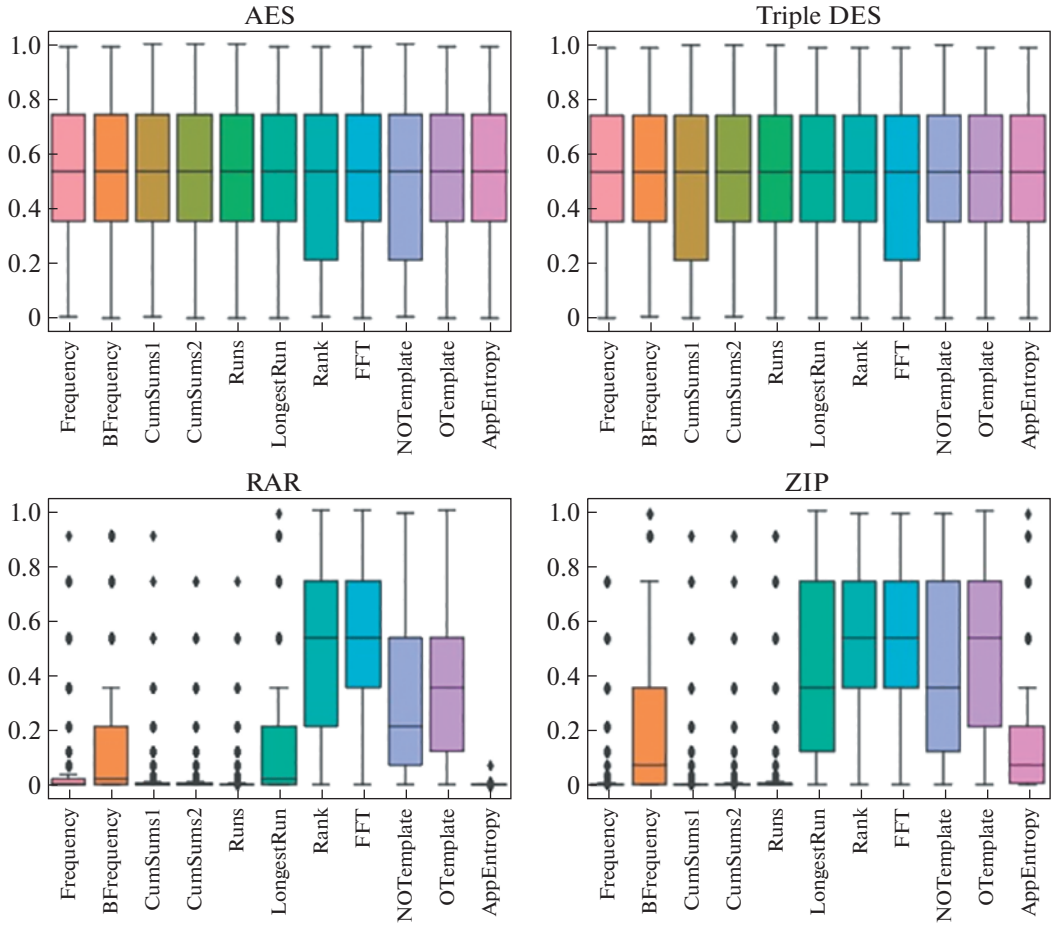


**Рис. 2.** Диаграммы размаха значений p-value тестов NIST для различных типов ПСП.

$$p - value = igamc\left(\frac{N}{2}, \frac{\chi^2_{obs}}{2}\right), \quad (10)$$

где *igamc* – неполная гамма-функция.

Неполная гамма-функция в общем виде определяется выражением (11):

$$Q(\alpha, x) = 1 - P(\alpha, x) = \frac{\Gamma(\alpha, x)}{\Gamma(\alpha)} = \frac{1}{\Gamma(\alpha)}\int_x^\infty e^{-t}t^{\alpha-1}dt, \quad (11)$$

где $Q(\alpha, 0) = 1$ и $Q(\alpha, \infty) = 0$.

Если значение p-value больше, чем выбранный порог $\alpha$, то принимается решение о подтверждении нулевой гипотезы и анализируемая последовательность считается случайной с уровнем доверия, определяемым выбранным пороговым значением $\alpha$. В противном случае нулевая гипотеза отвергается и последовательность считается неслучайной. Тест предназначен для тестирования последовательностей длиной не менее $10^6$ бит, что составляет примерно 122 кбайт. В описании теста указано, что длина шаблонов должна быть 9 или 10 бит, количество блоков длины $M$ бит $N \le 100$. Длина каждого блока задана в процедуре тестирования и равняется значению 131072. При использовании другого значения должны соблюдаться условия $M \ge 0.01 * n$ и $N = \left\lfloor \frac{n}{M} \right\rfloor$. Данные условия необходимы для обеспечения статистической значимости получаемых значений p-value.

В описании теста не указана причина использования подпоследовательностей длины 9 и 10 бит, кроме того, на их использование наложены достаточно жесткие ограничения в виде разделения подпоследовательности на определенное количество блоков определенной длины.

На основании данного теста была выдвинута гипотеза о возможности построения модели ПСП с использованием частот встречаемости подпоследовательностей ограниченной длины $N$ бит.

Для проверки гипотезы и выявления статистических особенностей в анализируемых ПСП были проведены эксперименты по подсчету частот встречаемости подпоследовательностей различной длины $N = [4, \ldots, 11]$ бит без перекрытия. При их обнаружении дальнейший подсчёт начинается

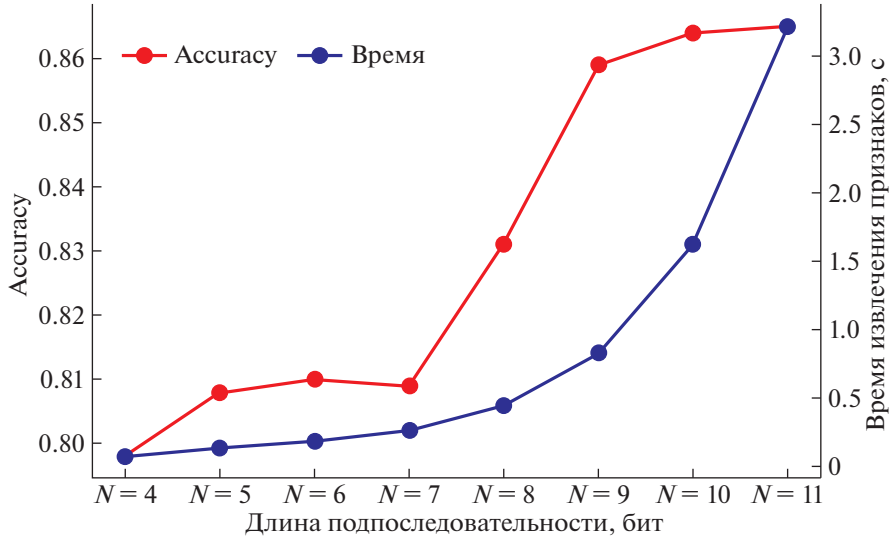


**Рис. 3.** Зависимость точности классификации ПСП алгоритмом построения случайного леса от длины подпоследовательностей, используемой моделью ПСП и времени извлечения признаков для одной последовательности.

со следующего бита после последнего бита подпоследовательности, если же совпадение не обнаружено, то происходит смещение на 1 бит. Подсчет частот встречаемости подпоследовательностей был выполнен согласно выражению (12):

$$f_j = F(j) = \frac{n(j)}{(M - N(j) + 1)}, \quad j \in \{0,\ldots,2^N\}, \quad (12)$$

где $f_j$ – частота встречаемости подпоследовательности $j$ в анализируемой ПСП; $n(j)$ – количество вхождений подпоследовательности $j$ в анализируемую ПСП; $M$ – длина анализируемой ПСП в битах; $N(j)$ – длина подпоследовательности $j$ в битах.

Данный подход, в отличие от статистических тестов NIST, где используются только непериодические шаблоны определенной длины, позволит проверить все возможные подпоследовательности и определить их дискриминирующую способность. Внедрение в разрабатываемую модель периодических последовательностей не достаточно обосновано с точки зрения статистических критериев на проверку случайности, однако имеет рациональное объяснение с точки зрения получения признаков, характеризующих конкретный класс ПСП.

Данный способ отличается от подсчета подпоследовательностей длиной 8 бит, так как получаемое распределение подпоследовательностей вычисляется по формуле (1), для распределения байт выполняется подсчет количества байт, и оно стремится к равномерному распределению.

Таким образом, модель ПСП представляет собой вектор статистических характеристик, определяемый выражением (13):

$$V_{sub} = (f_j,\ldots,f_{2^N}) \quad (13)$$

Для определения оптимальной длины подпоследовательности были проведены эксперименты, исходная выборка данных была преобразована в 8 наборов данных $\{V\} = \{V_4,\ldots,V_{11}\}$, содержащих в себе векторы ПСП, определяемые выражением (4), и состоящие из значения частот встречаемости подпоследовательностей длины 4–11 бит. Поскольку полученные значения частот встречаемости подпоследовательностей являлись малыми величинами (порядка $1 \times 10^{-5}$), то они были приведены к логарифмическому масштабу согласно выражению (4), что дает прирост точности классификации ПСП алгоритмами машинного обучения [37]:

$$V_{Sub}^{\ln} = \ln(V_{Sub}) = (\ln(f_j),\ldots,\ln(f_{2^N})). \quad (14)$$

Полученные признаковые пространства подавались на вход алгоритма построения случайного леса, результаты представлены на рисунке (3).

Наиболее рациональным значением длины подпоследовательностей, в зависимости от точности классификации и времени, затрачиваемого на извлечение частот подпоследовательностей, является значение 9 бит.

Отличительной особенностью полученного распределения 9-битных подпоследовательностей (рис. 4), нормированного по среднему значению частоты, является отсутствие его равномерности, что вводит новое вероятностное пространство, позволяющее моделировать ПСП.

Для проверки разработанной модели на адекватность также были проведены эксперименты по классификации ПСП различными алгоритма-

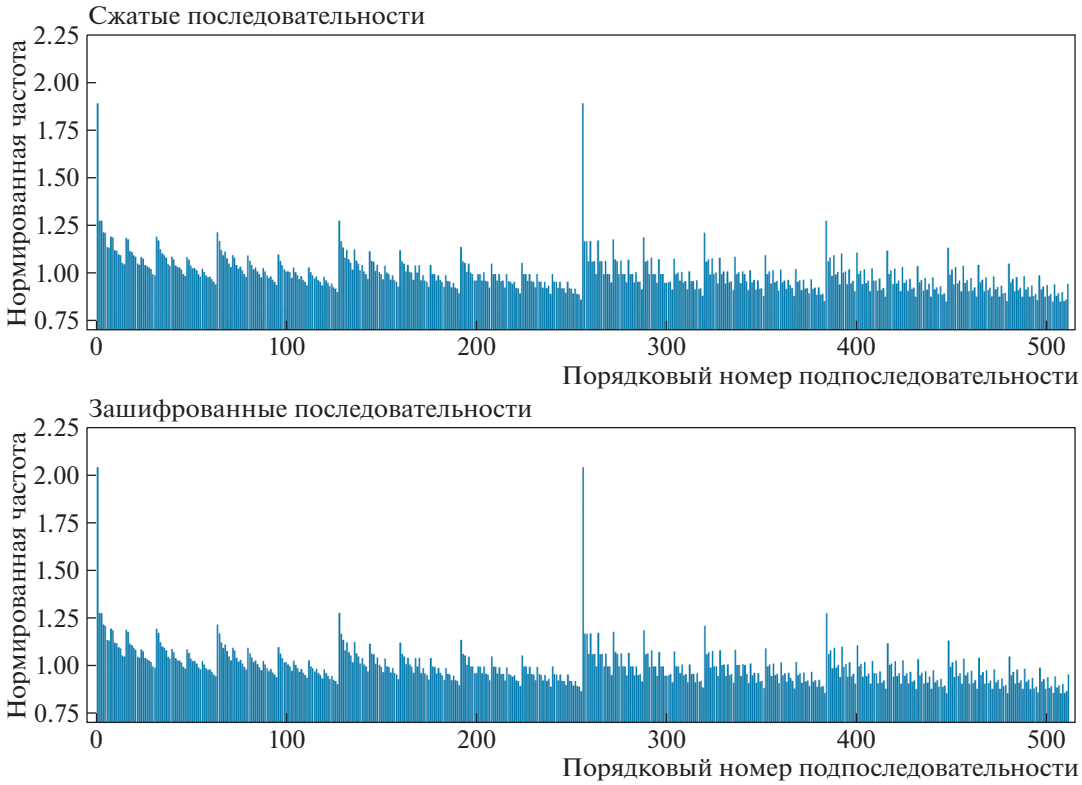


**Рис. 4.** Распределение 9-битных подпоследовательностей в ПСП.

ми машинного обучения, результаты представлены в таблице (5).

Полученные результаты свидетельствуют об ухудшении точности классификации алгоритмами построения случайного леса и дерева решений при использовании модели ПСП на основе учета частот встречаемости подпоследовательностей длины 9 бит, однако точность классификации алгоритмом $k$-ближайших соседей увеличилась на значение 0,012. Данный факт объясняется тем, что новая модель содержит в себе 512 значений вместо 256 при учете распределения байт. Алгоритмы построения случайного леса и дерева решений учитывают все имеющиеся в модели признаки, однако вероятно, не все из них имеют высокую дискриминирующую способность, из-за чего и происходит снижение точности классификации. Для алгоритма $k$-ближайших соседей наблюдается хоть и не значительная, но противоположная тенденция, так как данный алгоритм относится к метрическим и его точность возрастает с увеличением числа признаков.

Поскольку распределение байт и частоты подпоследовательностей длины 9 бит имеют различные распределения и вероятностные пространства, то наиболее рациональным шагом в проведении дальнейших исследований было построение синтезированной на их основе модели ПСП. К модели также были добавлены статистические признаки распределения байт: среднее значение ($B_{mean}$), среднеквадратическое отклонение ($B_{sko}$), минимальное ($b_{min}$) и максимальные ($b_{max}$) значения количества байт в ПСП, определяемые согласно выражению (15):

$$\begin{cases} B_{mean} = \dfrac{\sum n(b_i)}{256}, \\ B_{sko} = \sqrt{\dfrac{\sum (n(b_i) - B_{mean})^2}{256}}, \\ b_{min} = Min(n(b_i)), \\ b_{max} = Max(n(b_i)), \end{cases} \tag{15}$$

где $n(b_i)$ – количество появлений байта $i$ в анализируемой ПСП.

**Таблица 5.** Оценка точности классификации алгоритмами машинного обучения при использовании разработанной модели ПСП

| № п/п | Алгоритм | Accuracy |
|---|---|---|
| 1 | Random Forest | 0.859 |
| 2 | Decision Tree | 0.858 |
| 3 | kNN | 0.902 |

**Таблица 6.** Оценка точности классификации алгоритмами машинного обучения при использовании разработанной модели ПСП

| № п/п | Алгоритм | Accuracy |
|---|---|---|
| 1 | RF | 0.894 |
| 2 | DT | 0.891 |
| 3 | kNN | 0.91 |

**Таблица 7.** Оценка точности классификации алгоритмами машинного обучения при использовании разработанной модели ПСП и алгоритма классификации ПСП

| № п/п | Алгоритм | Accuracy |
|---|---|---|
| 1 | RF+DT | 0.97 |
| 2 | DT | 0.92 |
| 3 | kNN | 0.91 |

Таким образом, модель ПСП представляет собой вектор статистических характеристик, определяемый выражением (16):

$$V = (f_j, \ldots, f_{2^N}, b_0, \ldots, b_{255}, B_{mean}, B_{sko}, b_{min}, b_{max}), \quad (16)$$

Для оценки адекватности разработанной модели были проведены эксперименты по определению точности классификации ПСП алгоритмами машинного обучения, результаты представлены в таблице (6).

С помощью модели ПСП, учитывающей как равномерное распределение байт, так и новое распределение подпоследовательностей длины 9 бит, точность классификации ПСП увеличилась при использовании всех рассмотренных алгоритмов машинного обучения. Наивысшая точность наблюдалась у алгоритма kNN.

Однако полученные значения точности классификации ПСП не превышают значений при использовании модели ПСП на основе распределения байт. Кроме того, в настоящий момент модель ПСП представляет собой вектор статистических признаков длиной 772, что линейно влияет на время их извлечения и время выполнения классификации. Для преодоления указанных недостатков был разработан алгоритм классификации ПСП [38], учитывающий веса признаков, используемых в модели.

Оценка точности разработанного подхода представлена в таблице 7.

Для оценки влияния количества признаков на точность классификации были проведены эксперименты, результаты которых представлены на рисунке 5.

Наибольшая точность классификации была достигнута при использовании 110 признаков, однако точность классификации при 100 признаках меньше на 0.0001 пункт, а с ростом их количества в модели линейно увеличивается время выполнения процедуры извлечения признаков и классификации. Таким образом, на основе применения разработанного алгоритма классификации ПСП, учитывающего веса признаков и редуцирующего наименее значимые, был получен классификатор ПСП, достигающий точности классификации ПСП в 0.97.

На основании изложенного модель ПСП представляет собой вектор статистических характеристик согласно выражению (14).

Исследования в области информационной безопасности, связанные с применением алгоритмов машинного обучения также направлены

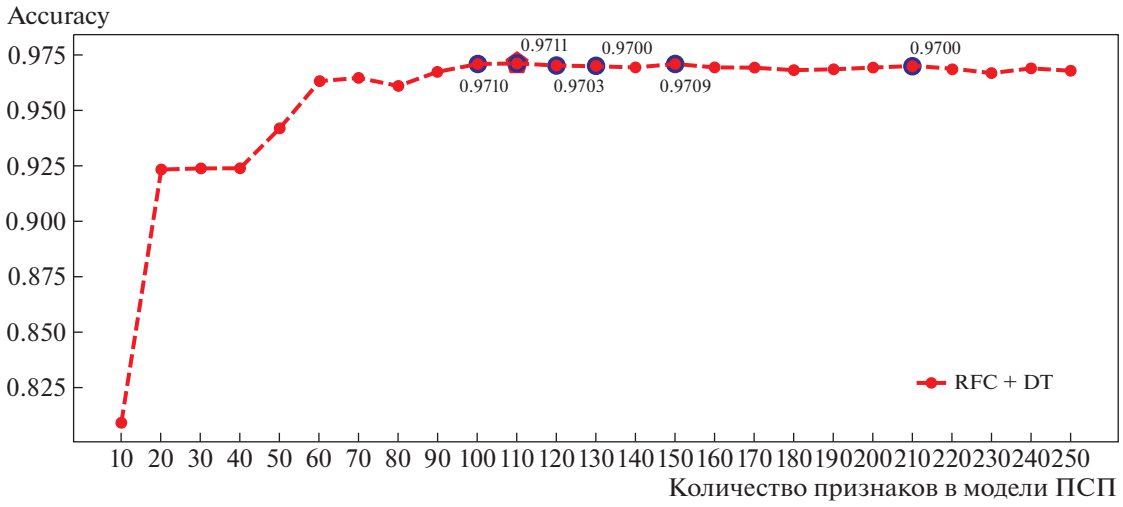


**Рис. 5.** Оценка влияния количества признаков в модели ПСП на точность классификации.

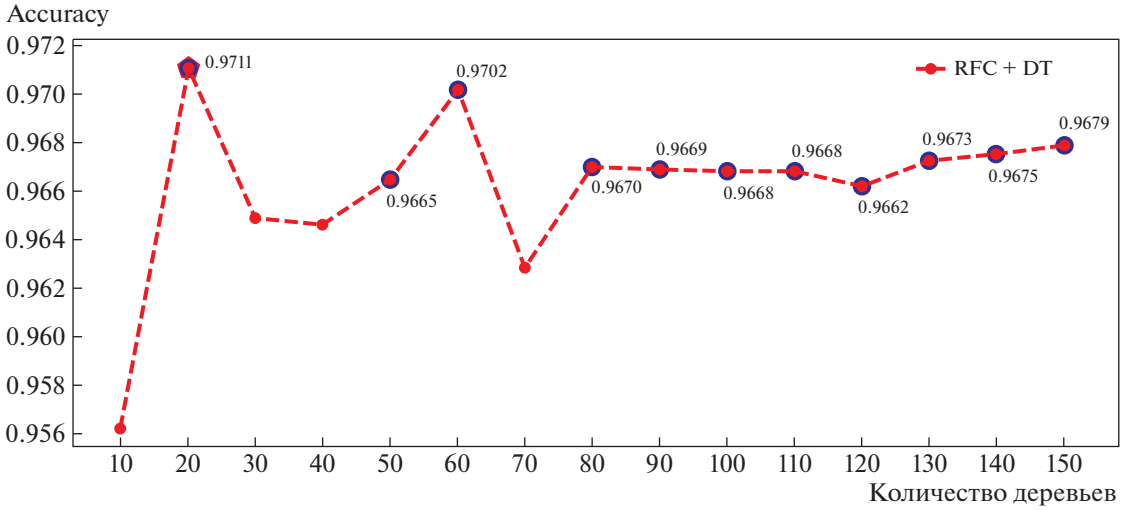


**Рис. 6.** Оценка влияния количества деревьев на точность классификации.

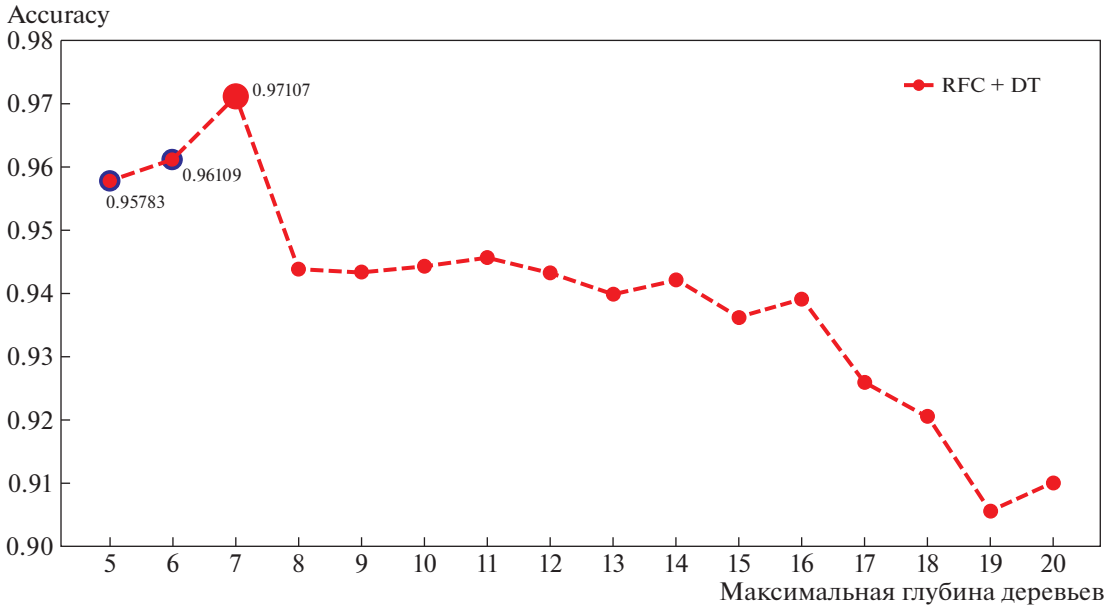


**Рис. 7.** Оценка влияния глубины деревьев на точность классификации.

на определение гиперпараметров используемых классификаторов [39]. Далее для достижения максимально возможной точности классификации ПСП необходимо установить параметры случайного леса и дерева решений.

С целью определения наиболее оптимальных параметров классификатора были проведены эксперименты по определению количества деревьев и их максимальной глубины: полученные результаты представлены на рисунках 6–7 соответственно.

Наиболее рациональным количеством деревьев в случайном лесу оказалось значение 20.

Наиболее рациональной глубиной деревьев в случайном лесу и дереве решений оказалось значение 7. Оно является локальным максимумом при незначительных колебаниях значений точности классификации и времени построения классификатора.

Также были проведены эксперименты по определению зависимости точности классификации от длины анализируемой последовательности. Результаты представлены на рисунке 8.

Высокой точности классификации удалось достичь при длине анализируемой последовательности в 600 кбайт, при ее увеличении с 50 кбайт

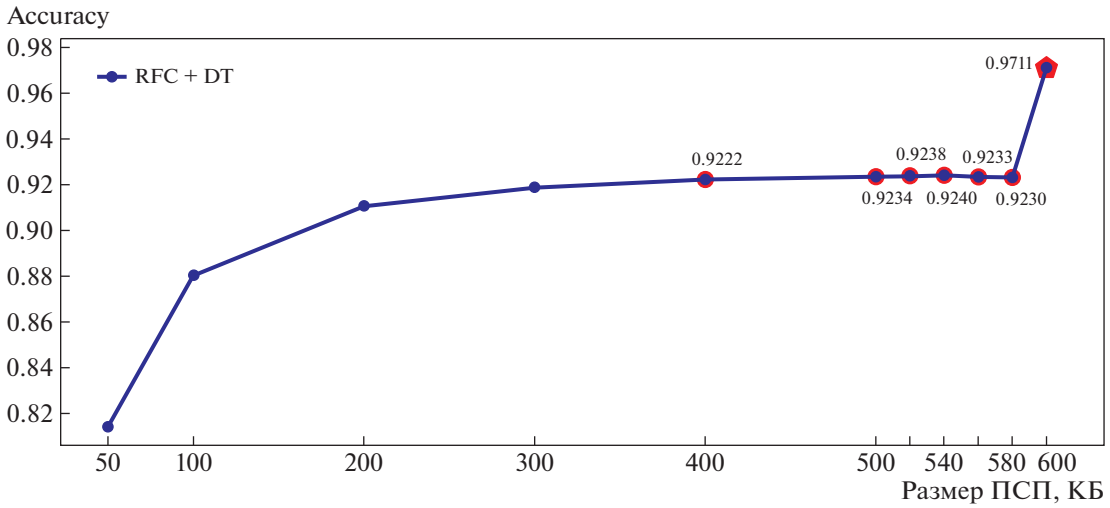


**Рис. 8.** Оценка точности классификации ПСП в зависимости от их длины.

увеличивается точность классификации. Данный факт объясняется возрастающими значениями частот подпоследовательностей и увеличением изменений в распределениях этих частот для зашифрованных и сжатых данных, что позволяет модели ПСП отражать статистические различия между ними.

## 4. ВЫВОДЫ

Основной вклад работы заключается в следующем:

1. Проведен анализ работ в области информационной безопасности на предмет использования методов классификации данных и алгоритмов машинного обучения. Сделан вывод о недостатках существующих подходов и предложены требования к разрабатываемому подходу классификации зашифрованных и сжатых данных до их передачи во внешнюю сеть.

2. Предложена модель ПСП, сформированных алгоритмами шифрования и сжатия данных (псевдослучайных последовательностей), отличающаяся от аналогов учетом распределения бинарных подпоследовательностей длины $N$ бит.

3. Сформированы ограничения, используемые в работе: для достижения максимальной точности классификации ПСП необходимы относительно большие фрагменты данных – не менее 600 кбайт; при использовании фрагментов около 50 кбайт точность по метрике, доля правильных ответов составляет 0.81. К достоинствам предложенного способа классификации ПСП следует отнести отсутствие учета в модели ПСП заголовков файлов и “магических” байт сжатых ПСП.

Разработанный подход показал высокую точность классификации зашифрованных и сжатых последовательностей, равную 0.97, и может быть применен для улучшения существующих DLP-систем или внедрен в сервер электронной почты для проведения процедуры анализа почтовых вложений перед их отправкой за периметр организации.

Исследование выполнено при финансовой поддержке Минобрнауки России (грант ИБ) в рамках научного проекта № 18/2020.

## СПИСОК ЛИТЕРАТУРЫ

1. *Le D.C., Zincir-Heywood N., Heywood M.I.* Analyzing data granularity levels for insider threat detection using machine learning // IEEE Transactions on Network and Service Management, 2020. V. 17. № 1. P. 30–44.
2. *Bhatiaa A., Bahugunaa A.A., Tiwaria K., Haribabua K., Vishwakarmab D.* A Survey on Analyzing Encrypted Network Traffic of Mobile Devices // arXiv preprint arXiv:2006.12352 [cs.CR]. 2020.
3. *Mamun M.S.I., Ghorbani A.A., Stakhanova N.* (2016) An Entropy Based Encrypted Traffic Classifier // In: Qing S., Okamoto E., Kim K., Liu D. (eds) Information and Communications Security. ICICS 2015. Lecture Notes in Computer Science. V. 9543. Springer, Cham. https://doi.org/10.1007/978-3-319-29814-6_23.
4. *Shen M., Wei M., Zhu L., Wang M.* Classification of encrypted traffic with second-order markov chains and application attribute bigrams // IEEE Transactions on Information Forensics and Security. 2017. V. 12. № 8. P. 1830–1843. https://doi.org/10.1109/TIFS.2017.2692682
5. *Zhang Z., Kang C., Fu P., Cao Z., Li Z., Xiong G.* Metric learning with statistical features for network traffic classification // IEEE 36th International Performance Computing and Communications Conference (IPCCC), San Diego, CA. 2017. P. 1–7. https://doi.org/10.1109/PCCC.2017.8280467.

6. *Yang Y., Kang C., Gou G., Li Z. Xiong G.*, TLS/SSL Encrypted Traffic Classification with Autoencoder and Convolutional Neural Network // IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Exeter, United Kingdom, 2018. P. 362–369. https://doi.org/10.1109/HPCC/SmartCity/DSS.2018.00079.
7. *Chen Y., Zang T., Zhang Y., Zhouz Y., Wang Y.* Rethinking Encrypted Traffic Classification: A Multi-Attribute Associated Fingerprint Approach // IEEE 27th International Conference on Network Protocols (ICNP), Chicago, IL, USA, 2019. P. 1–11. https://doi.org/10.1109/ICNP.2019.8888043.
8. *Wang P., Chen X., Ye F., Sun Z.* A survey of techniques for mobile service encrypted traffic classification using deep learning // IEEE Acces., 2019. V. 7. P. 54024–54033. https://doi.org/10.1109/ACCESS.2019.2912896
9. *Tang Z., Zeng X., Sheng Y.* Entropy-based feature extraction algorithm for encrypted and non-encrypted compressed traffic classification // International Journal of ICIC. 2019. V. 15. № 3. P. 845–860. https://doi.org/10.24507/ijicic.15.03.845
10. *Obasi T.C.* Encrypted Network Traffic Classification using Ensemble Learning Techniques // Doctoral dissertation, Carleton University, 2020. https://doi.org/10.22215/etd/2020-14171.
11. *Choudhury P., Kumar K.P., Nandi S., Athithan G.* An empirical approach towards characterization of encrypted and unencrypted VoIP traffic // Multimedia Tools and Applications. 2020. V. 79. № 1–2. P. 603–631. https://doi.org/10.1007/s11042-019-08088-w
12. *Yao Z., Ge J., Wu Y., Lin X., He R., Ma Y.* Encrypted traffic classification based on Gaussian mixture models and Hidden Markov Models // Journal of Network and Computer Applications. 2020. V. 166. P. 102711. https://doi.org/10.1016/j.jnca.2020.102711
13. *Baldini G., Hernandez-Ramos J.L., Nowak S., Neisse R., Nowak M.* Mitigation of Privacy Threats due to Encrypted Traffic Analysis through a Policy-Based Framework and MUD Profiles // Symmetry. 2020. V. 12. № 9. P. 1576. https://doi.org/10.3390/sym12091576
14. *Shen M., Liu Y., Zhu L., Xu K., Du X., Guizani N.* Optimizing Feature Selection for Efficient Encrypted Traffic Classification: A Systematic Approach // IEEE Network. 2020. V. 34. № 4. P. 20–27. https://doi.org/10.1109/MNET.011.1900366
15. *Panchenko A., Lanze F., Pennekamp J., Engel T., Zinnen A., Henze M., Wehrle K.* Website Fingerprinting at Internet Scale // Network and Distributed System Security Symp. 2016. P. 21–24. https://doi.org/10.14722/ndss.2016.23477
16. *Wei S., Ding Y., Han X.* TDSC: Two-stage DDoS detection and defense system based on clustering // In 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W). 2017. P. 101–102. https://doi.org/10.1109/DSN-W.2017.11.
17. *Sahoo K.S., Tripathy B.K., Naik K., Ramasubbareddy S., Balusamy B., Khari M., Burgos D.* An evolutionary SVM model for DDOS attack detection in software defined networks // IEEE Access. 2020. V. 8. P. 132502–132513. https://doi.org/10.1109/ACCESS.2020.3009733
18. *Grechishnikov E.V., Dobryshin M.M., Kochedykov S.S., Novoselcev V.I.* Algorithmic model of functioning of the system to detect and counter cyber attacks on virtual private network // Journal of Physics: Conference Series. 2019. V. 1203. № 1. P. 012064. https://doi.org/10.1088/1742-6596/1203/1/012064
19. *Добрышин М.М.* Предложение по совершенствованию систем противодействия DDoS-атакам // Телекоммуникации. 2018. № 10. С. 32–38. eLIBRARY ID: 36284311.
20. *Добрышин М.М., Спирин А.А., Лактионов А.Д.* Предложения по раннему обнаружению деструктивных воздействий Botnet на компьютерные сети связи. // Телекоммуникации. 2020. №. 12. С. 25–29. eLIBRARY ID: 44404522
21. *Zhu L., Tang X., Shen M., Du X., Guizani M.* Privacy-preserving DDoS attack detection using cross-domain traffic in software defined networks // IEEE Journal on Selected Areas in Communications. 2018. V. 36. № 3. P. 628–643. https://doi.org/10.1109/JSAC.2018.2815442
22. *Wang F., Quach T.T., Wheeler J., Aimone J.B., James, C.D.* Sparse coding for n-gram feature extraction and training for file fragment classification // IEEE Transactions on Information Forensics and Security. 2018. V. 13. № 10. P. 2553–2562. https://doi.org/10.1109/TIFS.2018.2823697
23. *Karampidis K., Papadourakis G.* File type identification-computational intelligence for digital forensics // Journal of Digital Forensics, Security and Law. 2017. V. 12. № 2. P. 6. https://doi.org/10.15394/jdfsl.2017.1472
24. *Karampidis K., Kavallieratou E., Papadourakis G.* Comparison of Classification Algorithms for File Type Detection. A Digital Forensics Perspective // Polybits. 2017. V. 56. P. 15–20. https://doi.org/10.17562/PB-56-2
25. *Kozachok A.V.* Development of a Heuristic Mechanism for Detection of Malware Programs Based on Hidden Markov Models // Automatic Control and Computer Sciences. 2018. V. 52. № 8. P. 1117–1123. https://doi.org/10.3103/S0146411618080345
26. *Srinivas M., Nayak A., Bhatt A.* Forged File Detection and Steganographic content Identification (FFDASCI) using Deep Learning Techniques // In CLEF (Working Notes). 2019. http://ceur-ws.org/Vol-2380/paper_142.pdf
27. *Konaray S.K., Toprak A., Pek G.M., Akçekoce H., Kılınç D.* Detecting File Types Using Machine Learning Algorithms // 2019 Innovations in Intelligent Systems and Applications Conference (ASYU). 2019. P. 1–4. https://doi.org/10.1109/ASYU48272.2019.8946393
28. *Casino F., Choo K.K.R., Patsakis C.* Hedge: Efficient traffic classification of encrypted and compressed packets // IEEE Transactions on Information Forensics and Security. 2019. V. 14. № 11. P. 2916–2926. https://doi.org/10.1109/TIFS.2019.2911156

29. *De Gaspari F., Hitaj D., Pagnotta G., De Carli L., Mancini L.V.* EnCoD: Distinguishing Compressed and Encrypted File Fragments // International Conference on Network and System Security, Springer, Cham. 2020. P. 42–62. https://doi.org/10.1007/978-3-030-65745-1_3.

30. *Mousavi S.S.* Detecting Disk Sectors Data Types Using Hidden Markov Model // 17th International ISC Conference on Information Security and Cryptology (ISCISC). 2020. P. 60–64. https://doi.org/10.1109/ISCISC51277.2020.9261906.

31. *Cheng L., Liu F., Yao D.* Enterprise data breach: causes, challenges, prevention, and future directions // Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 2017. V. 7. № 5. C. e1211.

32. *Doroud H.* et al. Speeding-up dpi traffic classification with chaining // IEEE Global Communications Conference (GLOBECOM). IEEE. 2018. C. 1–6.

33. *Hahn D., Apthorpe N., Feamster N.* Detecting compressed cleartext traffic from consumer internet of things devices // arXiv preprint arXiv:1805.02722. 2018.

34. *Wood D., Apthorpe N., Feamster N.* Cleartext data transmissions in consumer iot medical devices // Proceedings of the 2017 Workshop on Internet of Things Security and Privacy. 2017. C. 7–12.

35. *Scaife N., Carter H., Traynor P., Butler K. R.* Cryptolock (and drop it): stopping ransomware attacks on user data // IEEE 36th International Conference on Distributed Computing Systems (ICDCS). 2016. P. 303–312. https://doi.org/10.1109/ICDCS.2016.46.

36. *Raff E., Zak R., Cox R., Sylvester J., Yacci P., Ward R., Nicholas C.* An investigation of byte n-gram features for malware classification // Journal of Computer Virology and Hacking Techniques. 2018. V. 14. № 1. P. 1–20. https://doi.org/10.1007/s11416-016-0283-1

37. *Козачок А.В., Спирин А.А.* Алгоритм классификации псевдослучайных последовательностей // Вестник ВГУ. Серия: Системный анализ и информационные технологии. 2020. № 1. С. 87–98. https://doi.org/10.17308/sait.2020.1/2595

38. *Козачок А.В., Спирин А.А., Голембиовская О.М.* Алгоритм классификации псевдослучайных последовательностей на основе построения случайного леса // Доклады Томского государственного университета систем управления и радиоэлектроники. 2020. Т. 23. № 3. С. 55–60.

39. *Kozachok A.V., Kozachok V.I.* Construction and evaluation of the new heuristic malware detection mechanism based on executable files static analysis // Journal of Computer Virology and Hacking Techniques, 2018. V. 14. № 3. P. 225–231. https://doi.org/10.1007/s11416-017-0309-3

*ПРОГРАММИРОВАНИЕ, 2021, № 4, с. 45–55*

## ИНФОРМАЦИОННЫЙ ПОИСК

*УДК 004.9*

# МОДЕЛЬ И МЕТОД ОБНАРУЖЕНИЯ ИНФОРМАЦИОННЫХ КАМПАНИЙ

**© 2021 г. Д. Ю. Турдаков[a,b,*], С. В. Гарбук[c,**], П. В. Хенкин[d,***], И. С. Козлов[a,****], А. В. Лагута[a,*****], М. И. Варламов[a,******]**

[a] *Институт системного программирования им. В.П. Иванникова РАН*
*109004 Москва, ул. А. Солженицына, д. 25, Россия*

[b] *Московский государственный университет имени М.В. Ломоносова*
*119991 Москва, Ленинские горы, д. 1, Россия*

[c] *Национальный исследовательский университет "Высшая школа экономики"*
*101000 Москва, ул. Мясницкая, д. 20, Россия*

[d] *Фонд перспективных исследований*
*121059 Москва, Бережковская наб., д. 22, стр. 3, Россия*

**E-mail: turdakov@ispras.ru*

***E-mail: sgarbuk@hse.ru*

****E-mail: pkhenkin@yandex.r*

*****E-mail: kozlov-ilya@ispras.ru*

******E-mail: laguta@ispras.ru*

*******E-mail: varlamov@ispras.ru*

Поступила в редакцию 10.03.2021 г.
После доработки 15.03.2021 г.
Принята к публикации 19.03.2021 г.

Статья посвящена исследованию возможности автоматического выявления информационных кампаний в условиях отсутствия априорных знаний о факте проведения, целях, затрагиваемых объектах и целевой аудитории. В статье предлагается общая модель информационной кампании, а также выделяются признаки проведения скрытых информационных кампаний. Модель подходит для описания информационных кампаний как в социальных медиа, так и в традиционных СМИ, в том числе за пределами сети Интернет. На основе описанных признаков предложен метод обнаружения информационных кампаний, позволяющий решать задачу в автоматическом режиме.

Для подтверждения работоспособности метода было проведено экспериментальное исследование на данных, собранных из социальных медиа. Мы привлекли экспертов в смежных областях для разметки сообщений и создания тестового корпуса. С целью анализа сложности задачи мы оценили степень их согласия. Результаты анализа подтвердили первоначальную гипотезу, что даже для профессионалов, задача обнаружения скрытых информационных кампаний является нетривиальной. Тем не менее, используя метод голосования, мы построили тестовую коллекцию на которой провели исследование отдельных признаков, а также сравнения предложенного метода с отдельными ответами экспертов. Результат экспериментов подтвердил перспективность предложенного подхода к решению задачи обнаружения информационных кампаний.

**DOI:** 10.31857/S0132347421040063

## 1. ВВЕДЕНИЕ

Информационная кампания – это совокупность информационных сообщений, целенаправленно публикуемых в некотором промежутке времени, направленных на определенную целевую аудиторию, с целью побуждения этой аудитории к конкретным действиям.

Наиболее распространенными информационными кампаниями являются рекламные кампании, целью которых является повышение продаж товаров. Реклама может осуществляться *явно* в выделенных для этого блоках (отведенное время в сетке телевизионного вещания, рекламные блоки на веб-страницах) или *скрыто* с помощью более сложных техник, таких как продакт-плейсмент [1] и публикация заказных статей в СМИ. В отличие от рекламных, политические информационные кампании чаще используют скрытые методы

доведения информации до целевой аудитории. В работе [2] отмечается, что сообщения “в поддержку политического решения должны предоставлять информацию, помогающую целевой аудитории сделать свои собственные выводы”. В частности, информационные кампании являются эффективным инструментом для лоббизма политических решений [3].

В области исследования информационных войн, для обозначения схожего процесса используется термин “информационная операция” [4]. Информационные операции могут состоять из одной или нескольких повторяющихся информационных кампаний, между которыми делается пауза для сбора и анализа реакции целевого объекта [5]. Специфика информационных операций заключается в необходимости быть скрытыми, так как сам факт обнаружения информационной операции может помешать достижению ее цели. Актуальность проблемы выявления информационных операций отмечается в Стратегии научно-технологического развития Российской Федерации, утверждённой Указом Президента Российской Федерации от 1 декабря 2016 г. № 642, и Доктрине информационной безопасности Российской Федерации, утвержденной Указом Президента Российской Федерации от 5 декабря 2016 г. № 646.

Обнаружение скрытых информационных кампаний, а также анализ их целей, является важным шагом для понимания текущего состояния и динамики развития общества, без чего, в свою очередь, невозможно принятие эффективных стратегических решений, как на уровне бизнеса, так и уровне государственного управления.

В статье исследуются методы обнаружения скрытых информационных кампаний. Мы предлагаем формальную модель информационной кампании. Модель определяет стадии жизненного цикла информационных кампаний, для каждой стадии описываются косвенные признаки проведения информационных кампаний. Мы выбрали наиболее распространенные признаки и реализовали метод обнаружения скрытых информационных кампаний на их основе. Для подтверждения работоспособности представленного метода была разработана методика проведения экспериментального исследования. Основной проблемой при проведении экспериментального исследования оказалась сложность решения задачи обнаружения информационных кампаний для людей, даже для тех кто является экспертами в близких областях.

В следующей секции приведен обзор релевантных работ. Затем предлагается модель информационной кампании (секция 3) и признаки выявления скрытых информационных кампаний (секция 4). Далее описывается разработанный метод (секция 5). В секциях 6 и 7 описывается методика построения проверочного корпуса, анализируются ответы экспертов и приводятся результаты экспериментального исследования разработанного метода и отдельных признаков.

## 2. ОБЗОР РЕЛЕВАНТНЫХ РАБОТ

Наиболее релевантной работой среди отечественных авторов, является работа А.В. Потемкина [6]. Автор анализирует информационные операции в Интернет СМИ. Основная используемая гипотеза состоит в том, что информационные операции отличаются от естественного распространения тем, что делается информационный вброс на малоизвестных новостных ресурсах (“активная фаза”), а потом, после короткого затишья, новость появляется в более известных новостных ресурсах (“пассивная фаза”). Для естественного распространения – наоборот. Автор строит граф распространения новостей, где ребрами соединяются новости с похожим текстом. Похожесть считается методом шинглов. Направление задается временем опубликования. Новости группируются по сюжетам, объединяющим сообщения по ключевым словам в заданном промежутке времени. Далее ищутся шаблоны, удовлетворяющие основной гипотезе.

Наибольшее число статей в зарубежной литературе посвящено исследованию методов обнаружения информационных кампаний в социальных медиа. Под термином “социальные медиа” принято понимать социальные сети, форумы, блоги и другие сервисы, предоставляющие своим пользователям возможность взаимодействия друг с другом путём обмена сообщениями, комментирования, выставления оценок и др. Возрастающую роль социальных медиа, как средства воздействия на массовое сознание людей, отмечают многие исследователи в различных областях. Особенностями социальных медиа, в отличие от традиционных СМИ, являются простота публикации сообщений и моментальное доведение информации до целевой аудитории, независимо от границ государств. При этом информация, полученная от других людей в виде комментариев, сообщениях на форумах и в социальных сетях, постах в блогах, вызывает наибольшее доверие. В условиях отсутствия у пользователей инструментов проверки актуальности и достоверности полученной информации, социальные медиа открывают беспрецедентные возможности манипулирования людьми.

Наиболее релевантной нашему исследованию работой является статья [7] коллектива авторов из Техасского университета A&M и Университета штата Огайо США. Авторы подробно описывают метод поиска информационных кампаний в Twitter, основанный на группировке похожих сообщений и авторов. Изучаются методы анализа графов для выделения скоординированных кампа-

ний (coordinated campaigns) и методы вычисления схожести коротких текстовых сообщений.

Авторы работы [8] группируют сообщения в сюжеты с помощью алгоритма потоковой кластеризации, а затем предоставляют их на проверку эксперту. Авторы предполагают, что имея статистическую информацию о сюжете, представленном в виде дашборда, эксперт сможет сказать, является ли этот сюжет информационной кампанией или нет. Наше исследование показывает, что задача выявления информационных кампаний является сложной даже для экспертов и требует более продвинутых методов автоматизации.

Также стоит отметить работы, изучающие отдельные признаки информационных кампаний. Так в работе [9] авторы изучают поведение пользователей сети Twitter для выявления групп вредоносных ретвитеров. В работе [10] предлагается метод выявления информационных кампаний за счет выявления ботов влияния.

В этой работе мы ставим задачу разработки метода обнаружения информационных кампаний, который будет работать как для традиционных СМИ, так и для социальных медиа. Целенаправленное сокрытие осуществления информационных кампаний и огромный объем информационных потоков в социальных медиа делает разработку средств их обнаружения сложной теоретической и практической задачей.

## 3. МОДЕЛЬ ИНФОРМАЦИОННОЙ КАМПАНИИ

В наиболее широком понимании, **информационная кампания** – это любая *деятельность A* в *информационном пространстве F*, направленная на достижение *цели информационной кампании C*.

В общем случае *целью информационной кампании* является побуждение аудитории к определенному действию или бездействию [11]. Это может быть совершение определенной покупки, выход на площадь для протестов, или, наоборот, желание остаться дома для ограничения эпидемии. При этом особенность современной коммуникации такова, что призыв не будет выполнен аудиторией, если она к этому эмоционально не готова. Поэтому информационные кампании направлены именно на осуществления такой эмоциональной подготовки.

Таким образом, **целью информационной кампании** $C$ является создание у целевой аудитории $aud \subset U$ определенного эмоционального отношения к одному или нескольким целевым объектам $obj \in O$. Формально цель информационной кампании можно определить как множество

$$C = \{\{obj, s\}, aud\},$$

где $\{obj, s\} \subset \{O, S\}$ – множество пар объект–эмоциональная окраска. $U$ – множество пользователей. $O$ – множество целевых объектов. $S$ – множество типов эмоциональной окраски. Заметим, что в частном случае, целью информационной кампании может быть увеличение числа упоминаний объекта в информационном пространстве вне зависимости от эмоциональной окраски.

**Деятельность** $A$ по осуществлению информационной кампании заключается в создании информационных сообщений $M_{orig} = \{m_{orig}\} \subset M$, помогающих достичь цель, и поддержке распространения информации, для ее *доведения* до целевой аудитории и обеспечении ее *принятия*.

По аналогии с моделями информационных войн [11] возможно выделить следующие стадии жизненного цикла информационных кампаний:

1. Подготовка кампании. На этом шаге формируется план информационной кампании: анализируется отношение аудитории к целевым объектам, задается целевое отношение.

2. Подготовка инфраструктуры. Создается инфраструктура для распространения информации: создаются ресурсы для публикации материалов, на них привлекается целевая аудитория, производится “накрутка” популярности, в том числе создаются искусственные аккаунты (боты влияния). Так как создание новой инфраструктуры является сложной и дорогостоящей задачей, часто одна и та же инфраструктура используется для нескольких информационных кампаний.

3. Публикации первоначальной информации, создание информационного повода. Чаще всего информационный повод не создается “с нуля”, а ожидается подходящий информационный повод (реальное событие), информация о нем представляется в виде, способствующем достижению цели.

4. Поддержка доведения информации до целевой аудитории, с целью максимального распространения среди целевой аудитории. Сюда же отвлечение внимания аудитории на “более важные” события.

5. Анализ результатов информационной кампании и, при необходимости, повторение цикла.

Заметим, что для отдельно взятой информационной кампании обязательными являются только шаги 4 и 5. При этом, последний шаг может не оставлять наблюдаемых артефактов.

## 4. ПРИЗНАКИ ИНФОРМАЦИОННЫХ КАМПАНИЙ

Так как специфика изучаемых информационных кампаний предполагает скрытность, их выявление возможно только по косвенным признакам, нами разработан перечень признаков, проявляющихся при реализации информационных

**Таблица 1.** Признаки скрытых информационных кампаний на каждой стадии

| | |
|---|---|
| Подготовка кампании | Проведение социологических опросов;<br>Тестирование реакции небольшой фокус-группы на публикацию информации |
| Подготовка инфраструктуры | Признаки существующей инфраструктуры:<br>• использование ресурсов и аккаунтов в других, уже известных, информационных кампаниях;<br>• выявление контроля над ресурсами или аккаунтами организатора информационной кампании;<br>Признаки создания новой инфраструктуры:<br>• создание ресурсов (веб-сайтов, групп в социальных сетях и т.п.) для поддержки информационной кампании;<br>• создание искусственных аккаунтов, в т.ч. со специфическими характеристиками (пол, возраст, фотографии и т.п.);<br>• создание и искусственное увеличение аудитории и популярности ресурса (накрутка подписчиков, лайков и репостов);<br>• резкая смена тематики ресурса (при сохранении аудитории); |
| Публикация первоначальной информации и поддержка доведения информации (шаги 3, 4) | Дубликаты сообщений, направленных на изменения отношения к целевым объектам, в различных ресурсах от разных авторов;<br>Использование искусственных аккаунтов (ботов);<br>Всплеск числа негативных комментариев о целевых объектах;<br>Использование скомпрометированных ресурсов для публикации информации;<br>Ссылки на скомпрометированные ресурсы;<br>Использование характерных манипулятивных техник в тексте сообщений;<br>Публикация сообщений не соответствующих основной тематике ресурса;<br>Накрутка лайков для сообщений информационной кампании;<br>Массовые репосты и установка ссылок на сообщения информационной кампании. |
| Анализ результатов | Этап анализа результатов информационной кампании в общем случае не оставляет наблюдаемых артефактов, по которым можно было понять наличие и длительность этого этапа. Либо деятельность аналогична первому этапу |

кампаний различных типов и инвариантных к их конкретному содержанию (таблица 1). Приведенный список признаков не претендует на полноту. Однако заметим, что каждый элемент этого списка может быть обнаружен автоматизированными средствами.

Для полноты описания необходимо определить, что относится к использованию манипулятивных техник. Наиболее полный список перечислен в работе [12] и включает следующие техники: использование лжи, клеветы и дезинформации; провокация; аналитика и “квазианалитика”: статьи, оценивающие, интерпретирующие происходящие события, в т.ч. с учетом исторического контекста, направленные на изменение отношения к объекту; апеллирование к авторитету; подмена терминов; упрощенная “двуполярность” в интерпретации ситуации: «мы» и «враг»; использование специфичных дискурсивных конструкций: лозунги, агитация, пропаганда; акцентирование проблем; манипулирование ценностями; создание образа врага; поиск виновных; возложение вины на конкретную группу; формирование и развитие осознания идентичности; наставничество; двойные стандарты; создание коннотаций. Для определения некоторых из них предложены автоматические методы [13, 14]. Однако нам не известны работы по обнаружению большей части манипулятивных техник. Тем не менее, использование большинства из них может быть обнаружено с помощью методов машинного обучения.

Все признаки способны в той или иной мере определять информационные кампании на ранних стадиях. Однако признак на основе анализа всплеска числа негативных сообщений, в большинстве случаев, работает уже с реакцией на информационную кампанию, когда пользователи начинают выражать свой негатив по отношению к “информационному вбросу”. Таким образом способность признака определять информационную кампанию на ранней стадии ограничена. Тем не менее, в случае, если начальные сообщения информационной кампании выражают негатив по отношению к объекту мониторинга, они могут быть обнаружены с помощью данного подхода. Кроме того, признак на основе анализа всплеска числа негативных сообщений может быть полезен для увеличения полноты определения инфор-

мационных кампаний, если они не будут обнаружены на основе других признаков.

## 5. МЕТОД ОБНАРУЖЕНИЯ ИНФОРМАЦИОННЫХ КАМПАНИЙ

Мы ограничились обнаружением скрытых информационных кампаний на стадиях публикации первоначальной информации и поддержки доведения информации (стадии 3, 4). Нас интересовали признаки, инвариантные к конкретному содержанию информационной кампании (в частности, не использующие ключевые слова). Кроме того, в нашем исследовании мы ограничились только политической тематикой. Для этого был натренирован бинарный классификатор сообщений (логистическая регрессия), который позволил уменьшить объем данных для последующего анализа и увеличить их содержательность.

Предложенный метод обнаружения информационных кампаний состоит из следующих шагов:

1. Объединение отдельных сообщений в сюжеты;

2. Выявление признаков информационной кампании в сюжете.

Для объединения сообщений в сюжеты мы использовали метод, представленный в статье [15]. Метод состоит из двух шагов. На первом сообщения объединяются в кластеры на основе использования метода шинглов и “наивной кластеризации” по наличию общих специфичных именованны сущностей. На втором шаге содержимое кластеров уточняется, используя бинарный классификатор на парах сообщений.

Среди всех признаков мы выбрали пять наиболее распространенных (по нашим наблюдениям):

• Дубликаты сообщений, касающихся репутации целевых объектов, в различных ресурсах от разных авторов;

• Использование искусственных аккаунтов (ботов);

• Всплеск числа негативных комментариев о целевых объектах;

• Использование скомпрометированных ресурсов для публикации информации;

• Ссылки на скомпрометированные ресурсы.

Мы считали сообщения дубликатами, если их схожесть по мере Жаккара была больше 0.5. Для обнаружения таких сообщений мы воспользовались методом шинглов, по аналогии с алгоритмом выделения сюжетов. Для работы этого признака требуется только текст сообщений,однако полнота собираемых данных должна быть максимально высокой, так как сообщения-дубликаты публикуются в комментариях на различных ресурсах от имени разных людей. Обычный человек не может без специализированных инструментальных средств обнаружить такие публикации.

Для выявления искусственных аккаунтов использовался метод, описанный в работе [16]. Метод использует векторное представление вершин графа, для предсказания времени, через которое аккаунт может быть заблокирован. Мы считали, что признак сработал, если в дискуссии приняло участие не менее 3 ботов и доля сообщений от них среди всех сообщений была не менее 30%. Использование этого признака требует связи сообщений с аккаунтами авторов, а для работы используемого метода выявления ботов требуется граф социальных связей между этими аккаунтами.

В основе метода определения эмоционального отношения автора сообщения к упомянутому в этом сообщении объекту лежит модификация метода аспектно-ориентированного анализа эмоциональной окраски, предложенного на конференции SemEval-2016 [17]. Модификация заключается в удалении признаков, не релевантных для объектно-ориентированного анализа эмоциональной окраски (признаки, непосредственно связанные с аспектами).

Определение всплеска числа негативных комментариев о целевых объектах осуществлялось по следующему алгоритму:

1. Для каждого политического сообщения определяются объекты, упомянутые в сообщении и эмоциональное отношение к выявленным объектам. Для выявления объектов использовалась система Текстерра [18];

2. Сообщения группируются по времени написания (по часу);

3. Для каждого часа вычисляется число сообщений, написанных в этот час;

4. Для каждого объекта за каждый час вычисляется число сообщений, содержащих упоминания объекта;

5. Производится нормировка числа упоминаний: число упоминаний объекта делится на число сообщений за данный час

$$\frac{objectNum(hour)}{messageNum(hour)};$$

6. Для каждого объекта вычисляется среднее нормированное число упоминаний $m$ и стандартное отклонение нормированного числа упоминаний $\sigma$;

7. Число негативных сообщений по отношению к заданному объекту считается аномально большим если нормированное число упоминаний объекта превысило среднее более чем на 3 стандартных отклонения

$$num > m + 3 * \sigma.$$

Для получения списка скомпрометированных ресурсов брались ресурсы, в которых предложенным методом были не менее трех раз обнаружены сюжеты, содержащие другие признаки информа-

**Таблица 2.** Ответы экспертов при разметке корпуса для определения точности

| Номер эксперта | Номера сюжетов, отмеченные как информационные кампании |
|---|---|
| 1 | 1, 3, 5, 7, 8, 9, 10, 11, 13, 14, 15, 17, 19, 21, 22, 24, 25, 26, 27, 28, 33, 34, 35, 36, 37, 43, 44, 45, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 79, 81, 83, 84, 85, 87 |
| 2 | 1, 4, 5, 6, 8, 11, 13, 15, 39, 40, 42, 47, 70, 75, 77 |
| 3 | 1, 2, 3, 4, 5, 8, 10, 11, 13, 14, 15, 17, 19, 20, 21, 22, 24, 25, 26, 27, 29, 31, 33, 34, 36, 37, 38, 39, 40, 41, 45, 46, 48, 49, 52, 53, 54, 55, 59, 60, 62, 63, 64, 65, 67, 68, 69, 70, 71, 72, 73, 75, 76, 77, 79, 83, 84, 87 |
| 4 | 1, 2, 3, 4, 5, 8, 10, 11, 14, 17, 21, 22, 24, 26, 27, 29, 34, 35, 36, 37, 40, 42, 44, 46, 49, 51, 53, 54, 56, 57, 60, 62, 63, 64, 65, 67, 68, 69, 70, 73, 75, 81, 86 |
| 5 | 2, 3, 4, 5, 10, 11, 13, 17, 21, 24, 25, 27, 33, далее разметка не проводилась |
| 6 | 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 31, 33, 34, 36, 37, 38, 39, 40, 41, 42, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 67, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 81, 82, 83, 84, 86, 87 |
| 7 | 2, 8, 11, 13, 16, 23, 28, 33, 35, 59, 62, 64, 65, 67, 68, 75, 79, 83, 87 |

ционных кампаний. Публикация сообщений в этих ресурсах и ссылки из сообщений на эти ресурсы, в свою очередь, считались признаками информационных кампаний.

## 6. КОРПУС ТЕКСТОВ ДЛЯ ЭКСПЕРИМЕНТАЛЬНОГО ИССЛЕДОВАНИЯ

При проведении экспериментального исследования мы ограничились сообщениями социальной сети "Вконтакте[1]" и "Живого Журнала (ЖЖ)[2]". Были собраны все сообщения и комментарии с одного миллиона наиболее активных групп социальной сети Вконтакте и журналы 110 тысяч активных пользователей и сообществ ЖЖ за январь и февраль 2017 года. Суммарный объем собранных данных составил 168 Гб. Мы не собирали информацию с сайтов СМИ, так как многие из них имеют представительства в сети "Вконтакте", которые являются зеркалами основных сайтов. При этом пользователи могут оставлять комментарии к новостям. Таким образом, мы проанализировали существенный срез информационных потоков в сети Интернет.

Для измерения точности и полноты системы необходим корпус текстов, объединенных в сюжеты, в котором эксперты разметили эти сюжеты, на предмет, являются ли они информационными кампаниями. При этом, для измерения точности методов, в таком корпусе должно быть не менее нескольких десятков информационных кампаний. Однако, в реальности доля информационных кампаний среди всех сюжетов крайне мала, и необходимо разметить выборку из нескольких тысяч сюжетов, чтобы в итоговом корпусе набралось достаточное количество примеров. При этом, предполагается, что выявление информационной кампании является сложной для эксперта задачей, поэтому стандартный подход с разметкой большого корпуса в данном случае неприменим.

Для того, чтобы уменьшить объем работы экспертов было решено разметить два независимых корпуса, отдельно для *точности* и *полноты*. Для тестирования *полноты* экспертам был предложен список всех сюжетов по темам внешнеполитической деятельности РФ за 20–28 февраля 2017 года. Список содержал 70 сюжетов для разметки. Экспертам была поставлена задача выбрать, какие из этих сюжетов, по их мнению, имеют признаки информационной кампании. Для тестирования *точности* результатов была предложена случайная выборка сюжетов за февраль 2017 года, найденных автоматически различными методами. Выборка состояла из 87 сюжетов. Данные за январь использовались для составления базы скомпрометированных ресурсов.

Разметка корпусов производилась с 15 по 22 марта 2017 года. В разметке участвовало девять экспертов, им назначены номера 1–9 в описании экспериментов далее. С корпусом для тестирования *точности* работали эксперты 1–7, корпус для тестирования *полноты* размечали эксперты 2–9. Результаты разметки корпусов экспертами представлены в таблицах 2 и 3.

Эталонная выборка строится в зависимости от степени согласия экспертов: при высокой степени согласия в эталонную выборку попадают все сюжеты, которые эксперты посчитали информационной кампанией; при низкой степени согласия экспертов в выборку попадают сюжеты, которые заданная доля экспертов посчитала информационной кампанией. Так как определение информационной кампании – сложная для экс-

[1] https://vk.com
[2] https://www.livejournal.com

перта задача, мы ожидали, что степень согласия экспертов окажется низкой. В этом случае, необходимо исследовать зависимость точности и полноты результатов метода от числа согласившихся экспертов.

Возможны следующие случаи:

• Сюжет считается информационной кампанией, если хотя бы один эксперт посчитал его таковым (порог равен 0);

• Сюжет считается информационной кампанией, если все эксперты посчитали его таковым (порог равен 1);

• Сюжет считается информационной кампанией, если более одного эксперта посчитали его таковым (порог между 0 и 1).

Точность результатов работы метода высчитывается на эталонном корпусе для точности как

$$P = \frac{|S \cap E|}{|S|} = \frac{|E|}{|S|},$$

где $S$ – множество сюжетов–кандидатов, найденных методом и предложенных к разметке, $E$ – множество сюжетов–кандидатов, попавших в эталонную выборку. При таком определении точность может только уменьшаться при увеличении согласия экспертов, так как знаменатель не меняется, а в число сюжетов, на которых эксперты согласны, что это информационная кампания, может только уменьшиться при появлении новых экспертов.

Полнота результатов работы метода высчитывается на эталонном корпусе для полноты по формуле

$$R = \frac{|S \cap E|}{|E|}.$$

При такой постановке, полнота может как расти, так и уменьшаться в зависимости от порога согласия экспертов.

Для определения степени согласия экспертов был измерен коэффициент каппа Флейса (Fleiss' kappa), который позволяет определить согласие для любого фиксированного числа экспертов и любого числа оцениваемых объектов.

$$\kappa_F = \frac{\overline{P} - \overline{P_e}}{1 - \overline{P_e}},$$

где $1 - \overline{P_e}$ – степень согласия, достижимая случайно, $\overline{P} - \overline{P_e}$ – прирост в достигнутой степени согласия относительно случайного уровня.

На разметке данных для определения *точности* результатов метода каппа Флейса была равна $\kappa_F = 0.05$. Согласие при разметке корпуса для определения *полноты* – $\kappa_F = 0.22$.

**Таблица 3.** Ответы экспертов при разметке корпуса для определения полноты

| Номер эксперта | Номера сюжетов, отмеченные как информационные кампании |
|---|---|
| 4 | 3, 5, 13, 14, 16, 17, 20, 31, 39, 43, 46, 55, 69 |
| 9 | 1 |
| 5 | 2, 3, 5, 6, 17, 18, 23, 31, 35, далее разметка не проводилась |
| 2 | 31, 34 |
| 8 | 1 |
| 7 | 3, 4, 13, 31, 34, 39, 43, 46, 66, 69, 70 |
| 6 | 2, 3, 4, 5, 6, 13, 15, 16, 23, 35, 38, 43, 46, 59, 66, 69, 70 |
| 3 | 2, 3, 5, 6, 13, 14, 16, 23, 31, 43, 46, 66, 69, 70 |

**Таблица 4.** Референсные значения коэффициента каппа Флейса

| $\kappa_F$ | Интерпретация |
|---|---|
| <0 | Отсутствие согласия |
| 0.01–0.20 | Крайне низкое согласие |
| 0.21–0.40 | Низкое согласие |
| 0.41–0.60 | Умеренное согласие |
| 0.61–0.80 | Существенное согласие |
| 0.81–1.00 | Почти полное согласие |

Сравнение с референсными значениями каппы Флейса (таблица 4) показывает, что согласие экспертов при разметке было *крайне низким* при разметке корпуса для определения точности и *низким* при разметке корпуса для определения полноты, что говорит о сложности задачи определения информационных кампаний для человека.

Для более детального понимания проблемы было изучено попарное согласие экспертов. Результаты представлены на рисунках 1–4. Для измерения согласия использовались следующие меры:

• Коэффициент каппа Коэна $\kappa_C(A, B) = \frac{p_0 - p_e}{1 - p_e}$,

где $p_0$ – относительное наблюдаемое согласие между двумя экспертами, $p_e$ – вероятность случайного согласия экспертов.

• Мера Жаккара $Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|}$

• Точность $Precision(A, B) = \frac{|A \cap B|}{|A|}$

• Полнота $Recall(A, B) = \frac{|A \cap B|}{|B|}$, где $A$ и $B$ – сюжеты, отмеченные первым и вторым эксперта-

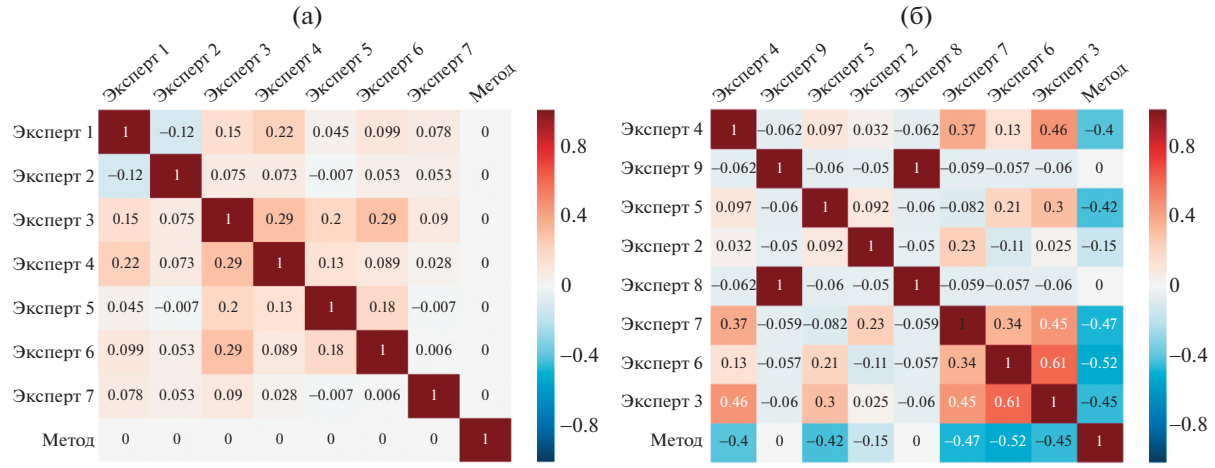


**Рис. 1.** Попарное согласие экспертов (и метода) по коэффициенту **каппа Коэна** при разметке данных для определения **точности** (а) и **полноты** (б) результатов метода.

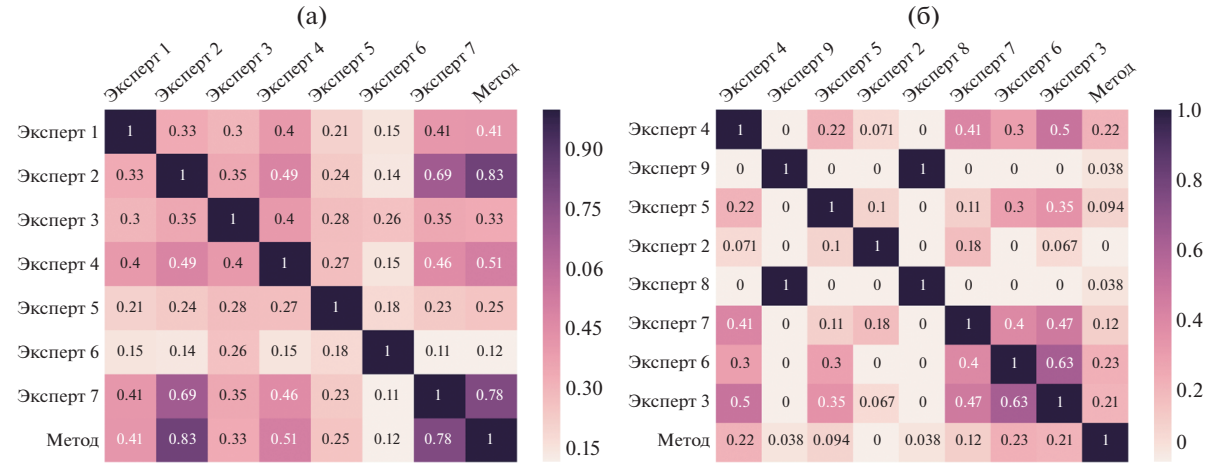


**Рис. 2.** Попарное согласие экспертов (и метода) по мере **Жаккара** при разметке данных для определения **точности** (а) и **полноты** (б) результатов метода.

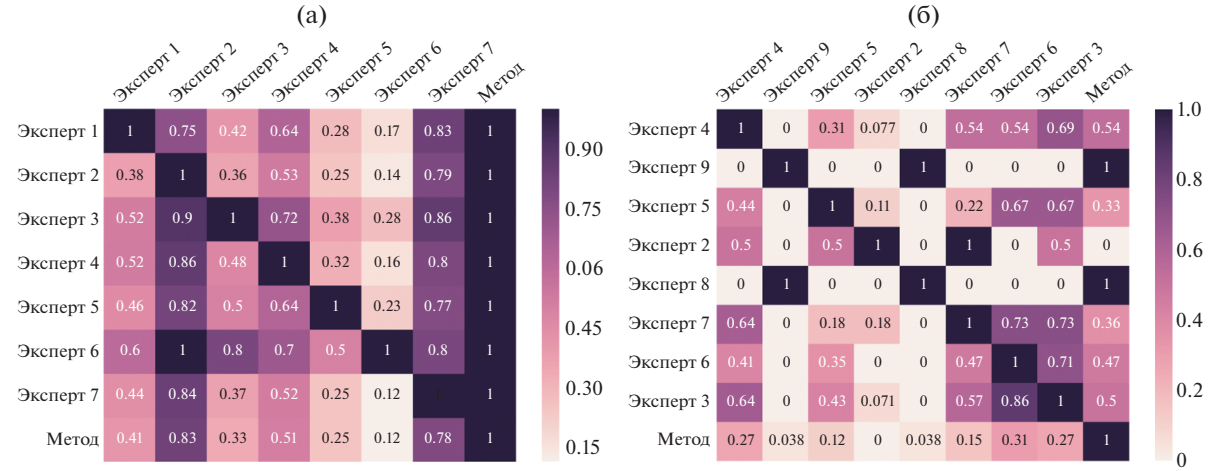


**Рис. 3.** Попарное согласие экспертов (и метода) по мере **Точности** при разметке данных для определения **точности** (а) и **полноты** (б) результатов метода.

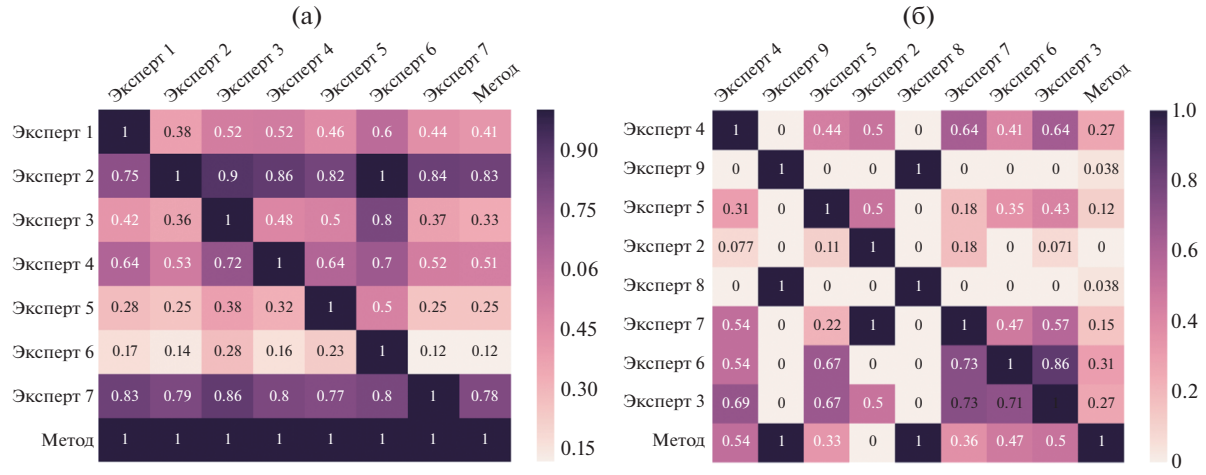


**Рис. 4.** Попарное согласие экспертов (и метода) по мере **Полноты** при разметке данных для определения **точности** (а) и **полноты** (б) результатов метода.

ми соответственно, как информационные кампании.

На рисунке 1 представлено попарное согласие экспертов (и предлагаемого метода) по коэффициенту каппа Коэна при разметке данных для определения точности (а) и полноты (б).

Каппа Коэна для предложенного метода и любого эксперта на эталонной коллекции для измерения точности всегда равна нулю. Так получается из-за того, что эта коллекция была сформирована на основе ответов, выдаваемых разработанным методом, поэтому наблюдаемое согласие с ним не отличается от случайного.

Для интерпретации остальных значений можно воспользоваться таблицей 4. В большинстве случаев попарное согласие экспертов является низким, что характеризует задачу выявления информационных кампаний, как крайне сложную для экспертов.

Стоит также отметить полное согласие двух экспертов (8 и 9) при разметке данных для определения **полноты**. Однако если обратиться к таблице 3, содержащей ответы экспертов, то можно увидеть, что оба эксперта отметили только первый сюжет, как информационную кампанию. При этом другие эксперты не посчитали этот сюжет содержащим признаки информационных кампаний. Более того в разметке точности упомянутые эксперты не участвовали. Исходя из этого можно сделать предположение, что эксперты только ознакомились с системой разметки, однако саму разметку не производили. В связи с этим, было решено убрать их ответы из эталонного корпуса. Это увеличило согласованность экспертов при разметке данных для определения полноты: $\kappa_F = 0.35$. Тем не менее, согласие экспертов осталось *низким*.

## 7. ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ МЕТОДА

На рисунке 5 представлены значения точности (а) и полноты (б) ответов экспертов и метода на соответствующих эталонных корпусах, в зависимости от значения порога согласия экспертов: в эталонный корпус попадали ответы, если с ним было согласно не менее порогового числа экспертов. Точность в таблице 5а вычислялась как

$$P = \frac{|S \cap E|}{|S|},$$

где $S$ – множество ответов эксперта (или системы), $E$ – множество информационных кампаний в эталонной выборке.

В разметке данных для определения *точности* участвовало 7 экспертов. Поэтому рисунок 5а содержит 7 столбцов. В разметке данных для определения *полноты* участвовало 6 экспертов (те же за исключением 1), однако не оказалось ни одного сюжета, где бы все 6 экспертов согласились, что он является информационной кампанией. Поэтому на рисунке 5б представлено только 5 столбцов.

Точность и полнота ответов системы на эталонных корпусах представлена в последней строке рисунков 5а и 5б соответственно.

На рисунке 6 показаны значения точности (а) и полноты (б) отдельных признаков на соответствующих эталонных корпусах, полученных при заданном значении порога согласия экспертов.

- *dublicates* – дубликаты сообщений;
- *badSources* – публикация в скомпрометированных источниках;
- *botComments* – участие ботов;
- *negative* – всплеск числа негативных сообщений;

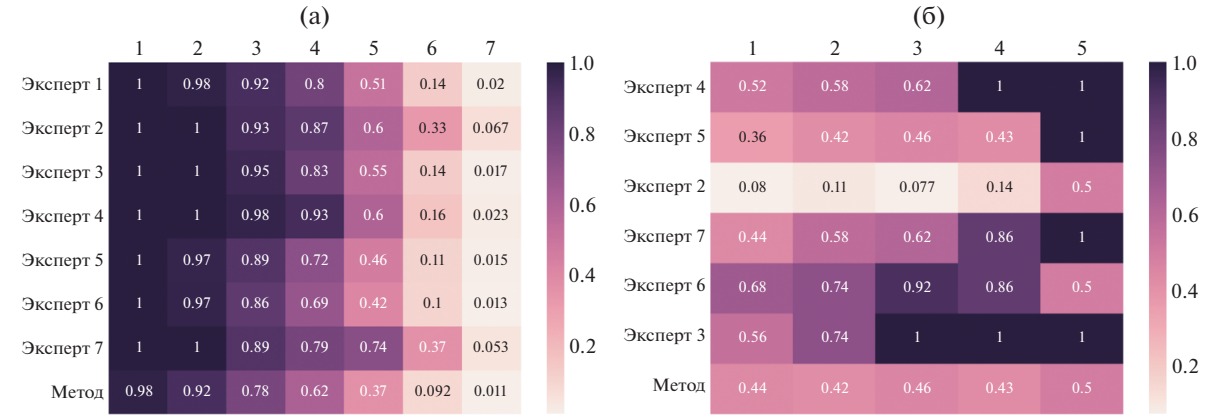


**Рис. 5.** Зависимость точности (а) и полноты (б) ответов экспертов и метода на соответствующих эталонных корпусах, полученных при заданном значении порога согласия экспертов.

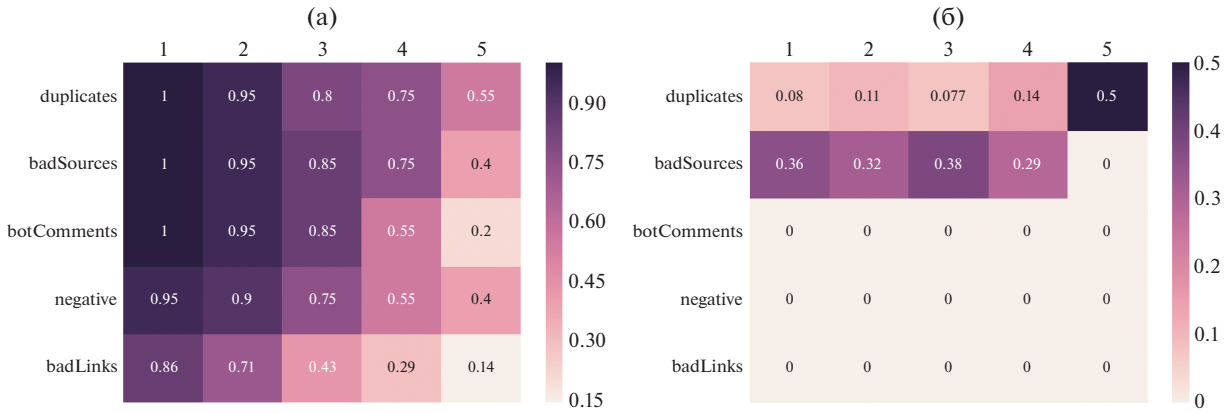


**Рис. 6.** Зависимость точности (а) и полноты (б) методов на эталонных корпусах, полученных при заданном значении порога согласия экспертов.

• *badLinks* – ссылки на скомпрометированные источники.

Лучшую точность показывают методы на основе поиска дубликатов и на основе недостоверных источников. В эталонный корпус для проверки *полноты* не попали информационные кампании, обнаруживаемые тремя последними методами, поэтому значения полноты нулевые. Исходя из этого можно сделать вывод, что улучшение методов выявления и анализ инфраструктуры распространения информации является перспективным направлением для исследований.

## 8. ЗАКЛЮЧЕНИЕ

В работе мы предложили формализацию понятия “информационная кампания”. Было выделено пять этапов проведения информационной кампании и для каждого из этапов определены признаки, которые позволяют обнаруживать скрытые информационные кампании. На основе наиболее распространенных из этих признаков был реализован метод выявления информационных кампаний, позволяющий получать результат в автоматическом режиме независимо от содержания кампании.

Особое внимание было уделено проведению экспериментального исследования предложенного метода. Задача выявления информационных кампаний оказалась крайне сложной для людей, о чем свидетельствует низкая степень согласованности ответов экспертов при разметке эталонных корпусов. В связи со сложностью получения эталонных данных, мы разделили задачу на две части, и отдельно измерили точность и полноту. Кроме того мы измерили точность и полноту метода и отдельных признаков в зависимости от порога согласия экспертов. Результаты измерений показали, что разработанный метод сравним по качеству с экспертной оценкой, однако сами признаки не обладают достаточной полнотой для их использования отдельно от других. Таким об-

разом, расширение предложенного метода путем добавления алгоритмов автоматического выявления остальных описанных в работе признаков является перспективным направлением работы.

## СПИСОК ЛИТЕРАТУРЫ


1. *Березкина О.П.* Product Placement: технология скрытой рекламы. Издательский дом “Питер”, 2008.
2. *Годдард Б.* Кампании поддержки политических решений. Справочник по политическому консультированию / под ред. Д.Д. Перлматтера, 2002.
3. *Павроз А.В.* Информационные кампании в современном лоббизме. Вестник Пермского университета. Серия: Политология, 2014. № 2. С. 66–74.
4. *Расторгуев С.П.* Планирование и моделирование информационной операции. Информационные войны, 2014. № 1. С. 2–10.
5. *Манойло А.В.* “Дело Скрипалей” как операция информационной войны // Вестник Московского государственного областного университета, 2019. № 1.
6. *Потемкин А.* Распознавание информационных операций средств массовой информации сети интернет. Интернет-журнал Науковедение. 2015. Т. 3. № 28. С. 14.
7. *Lee K., Caverlee J., Cheng Z., Sui D.Z.* Campaign extraction from social media // ACM Trans. Intell. Syst. Technol. 2014. V. 5. № 1. P. 9:1–9:28. https://doi.org/10.1145/2542182.2542191
8. *Assenmacher D., Clever L., Pohl J.S., Trautmann H., Grimme C.* A two-phase framework for detecting manipulation campaigns in social media // International Conference on Human-Computer Interaction, Springer, 2020. P. 201–214.
9. *Vo N., Lee K., Cao C., Tran T., Choi H.* Revealing and detecting malicious retweeter groups. В 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), IEEE, 2017. P. 363–368.
10. *Abu-El-Rub N., Mueen A.* Botcamp: bot-driven interactions in social campaigns // The World Wide Web Conference, 2019. P. 2529–2535.
11. *Неждановв И.Ю.* Технологии информационных войн в интернете. [PDF] http://bash.rosmu.ru/activity/attach/events/1283/01.pdf, 2001.
12. *Кара-Мурза С.* Манипуляция сознанием. Век XXI. 2017.
13. *Zhou X., Zafarani R.* Fake news: a survey of research, detection methods, and opportunities. arXiv preprint arXiv:1812.00315, 2, 2018.
14. *Snefjella B., Lana N., Kuperman V.* How emotion is learned: semantic learning of novel words in emotional contexts // Journal of Memory and Language. 2020. V. 115. P. 104171.
15. *Скорняков К.А., Ласкина А.С., Турдаков Д.Ю.* Двухшаговый метод объединения новостей в сюжеты // Труды Института системного программирования РАН. 2020. Т. 32. № 4.
16. *Skorniakov K., Turdakov D., Zhabotinsky A.* Make social networks clean again: graph embedding and stacking classifiers for bot detection // Proceedings of the 27th ACM International Conference on Information and Knowledge Management 2018.
17. *Mayorov V., Andrianov I.* Mayand at semeval-2016 task 5: syntactic and word2vec-based approach to aspect-based polarity detection in russian. В Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 325–329, San Diego, California. Association for Computational Linguistics, 2016.
18. *Турдаков Д., Астраханцев Н., Недумов Я., Сысоев А., Андрианов И., Майоров В., Федоренко Д., Коршунов А., Кузнецов С.* Texterra: инфраструктура для анализа текстов. Труды Института системного программирования РАН, 2014. Т. 26. № 1.

*ПРОГРАММИРОВАНИЕ, 2021, № 4, с. 56–62*

## КОМПЬЮТЕРНАЯ ГРАФИКА И ВИЗУАЛИЗАЦИЯ

*УДК 004.421.6*

# УЛУЧШЕНИЕ СЕГМЕНТАЦИИ ПАТОЛОГИЙ ЛЕГКИХ И ПЛЕВРАЛЬНОГО ВЫПОТА НА КТ-СНИМКАХ ПАЦИЕНТОВ С COVID-19

**© 2021 г. Д. С. Лащенова[a],*, А. М. Громов[b],**, А. С. Конушин[a],***, А. М. Мещерякова[b],******

*[a] Московский государственный университет имени М.В. Ломоносова, 119991 Москва, Ленинские горы, д. 1, Россия*

*[b] ООО "Платформа Третье Мнение", 121205 Москва, территория Сколково инновационного центра, ул. Нобеля, д. 7, Россия*

**E-mail: daria.laschenova@graphics.cs.msu.ru*

***E-mail: alexander.gromov@3opinion.ai*

****E-mail: anton.konushin@graphics.cs.msu.ru*

*****E-mail: ceo@3opinion.ai*

Поступила в редакцию 10.10.2020 г.
После доработки 20.10.2020 г.
Принята к публикации 12.01.2021 г.

В 2020 пандемия коронавируса затронула миллиарды людей по всему свету и заставила пересмотреть отношение к системам здравоохранения и к методам, используемым в современной медицине. Ввиду высокой нагрузки на радиологов и врачей появилась необходимость автоматических систем выявления патологий на медицинских исследованиях. Множество работ, посвященных работе с КТ-снимками пациентов с Covid-19, предполагают внедрение в системы медицинской помощи. Но улучшение по "классическим" метрикам вроде mAP или IoU по всем исследованиям не всегда отображает улучшение модели с точки зрения врачей. В данной работе было предложено считать метрики, усредняя не по всем исследованиям, а по группам в зависимости от размера патологий, а также оценивать количество ложноположительных участков найденных вне легких, поскольку наличие таких участков очень негативно оценивается врачами. Так же был предложен метод, улучшающий сегментацию патологий легких и плеврального выпота, с учетом замечаний, которые были высказаны выше.

**DOI:** 10.31857/S0132347421030067

## 1. ВВЕДЕНИЕ

Covid-19 – болезнь, вызываемая вирусом SARS-CoV-2. Часто встречающееся осложнение при болезни – вирусная пневмония.

Ранняя диагностика осложнений может уменьшить время и интенсивность лечения. Обычно для нее используется компьютерная томография. КТ-исследование – это радиологическое трехмерное исследование части тела, сконструированное компьютером из последовательности плоскостных поперечных срезов сделанных вдоль одной оси. Несмотря на то, что КТ-исследование не является основным в диагностике Covid-19, оно позволяет осуществлять диагностику осложнений, приоритизацию пациентов по степени тяжести осложнений, оценивать динамику болезни. Радиологи находят на исследовании различные патологии, оценивают их тип и размер. Затем эта информация используется для того, чтобы оценить состояние пациента и назначить лечение. Выделяют 5 степеней тяжести пациента: КТ-0, КТ-1, КТ-2, КТ-3, КТ-4. Они определяются по доле объема пораженных легких. Это позволяет назначить лечение и понять, нужна ли госпитализация пациенту. В редких случаях при тяжелой степени появляется плевральный выпот – избыточное скопление жидкости в плевральной полости. Для врачей важно обнаружить данную патологию и оценить объем жидкости и ее тип.

Целью этого исследования является улучшение системы, автоматически сегментирующей легкие, патологии и плевральный выпот на КТ-снимке, чтобы можно было оценить процент поражения легких и объем плеврального выпота.

## 2. ОБЗОР ОБЛАСТИ

Поскольку Covid-19 стремительно распространился и еще продолжает распространяться, специалисты в компьютерном зрении начали поиск

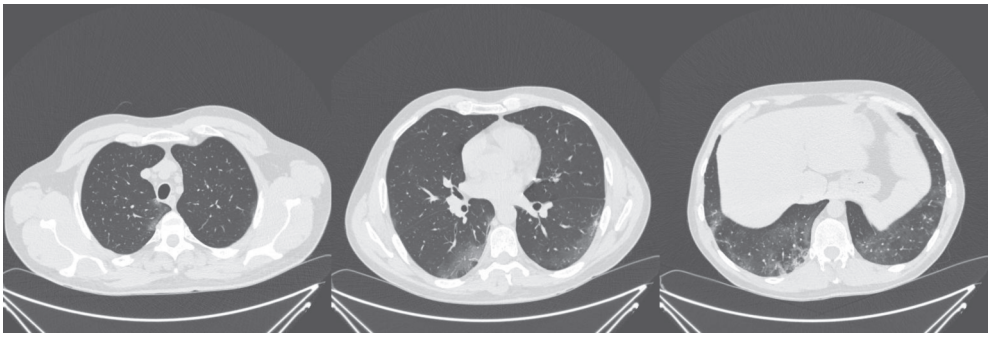

**Рис. 1.** Пример трех срезов из одного КТ-исследования.

решений, которые могут помочь радиологам в их работе.

В самом начале, до массового производства тестов на вирус SARS-CoV-2, было важно предсказать, насколько вероятно, что эта пневмония вызвана именно коронавирусом. В связи с этим появлялось много работ по задаче классификации: Linda Wang [1] представила сеть COVID-Net, которая определяла, болен ли пациент и болен ли он Covid-19. Lin Li предложил сеть COVNet [2] для решения аналогичной задачи. Но подобные решения потеряли релевантность в связи с тем, что доступность теста на вирус стала выше, чем возможностьс делать КТ-исследование.

Для того, чтобы систему можно было использовать для задач маршрутизации и отслеживания динамики состояния пациента, необходимо, чтобы она решала задачу сегментации патологий и определенных типов патологий. Fei Shan редложил 3D-модель VB-Net [3] для сегментации патологий, долей легкого и сегментов легкого, используя стратегию human-in-the-loop. Parham Yazdekhasty, Ali Zindar [4] используют U-Net-подобную сеть, но используют два декодера для предсказания класса легких и класса патологий, а затем объединяют результаты для более качественного поиска патологоий. Кто-то объединяет задачи классификации и сегментации, как, например, Amine Amyar [5].

## 3. ВЫБОРКА

Выборка была предоставлена компанией ООО “Платформа Третье Мнение” [6]. Она содержит 938 КТ-исследований легких пациентов с Covid-19. Исследования были получены в формате dicom. Предварительная обработка данных не производилась. Примеры срезов можно увидеть на рис. 1.

10–20 срезов из каждого исследования были отданы на разметку радиологам, что дало 18673 изображения со срезами легких. Радиологи с помощью полигонов выделили на них области, содержащие легкие, патологии легких и плевральный выпот.

Выборка была разделена на 2 части: тренировочную и тестовую, содержащие 85 и 15% соответственно.

Стоит отметить, что радиологи обычно рисуют полигоны с более сглаженными границами, а модель обычно дает более точные границы. Также некоторые патологии имеют размытые границы на изображении, что усложняет построение точной маски. Это приводит к тому, что значения метрик будут не близки к идеальным.

## 4. МЕТРИКИ

### *4.1. Общие метрики*

В качестве метрик используются IoU и среднее AP по исследованиям.

Полнота определяется как количество верно определенных положительных пикселей, деленное на общее количество положительных пикселей. Точность определяется как количество верно определенных положительных пикселей деленное на общее количество пикселей, отмеченных, как положительные. AP вычисляется как площадь под графиком точность–полнота, где каждая точка графика показывает значение точности и полноты при различных порогах, по которым отсекаются пиксели, определенные моделью как положительные.

IoU показывает, насколько сильно пересекается маска в разметке и полученная моделью:

$$IoU = \frac{TP}{FN + TP + FP}$$

Стоит также заметить, что при выбранных метриках ошибка в малом количестве пикселей может приводить к различному падению метрик в зависимости от того, сколько истинных пикселей было в размеченной маске. Сложная ситуация возникает с исследованиями, на которых нет пикселей определенного класса. По ним нельзя вычислить mAP. Если пикселей этого класса не нашлось моделью, то не вычисляется и IoU (или его можно принять 1). Если же нашелся хотя бы один, то IoU

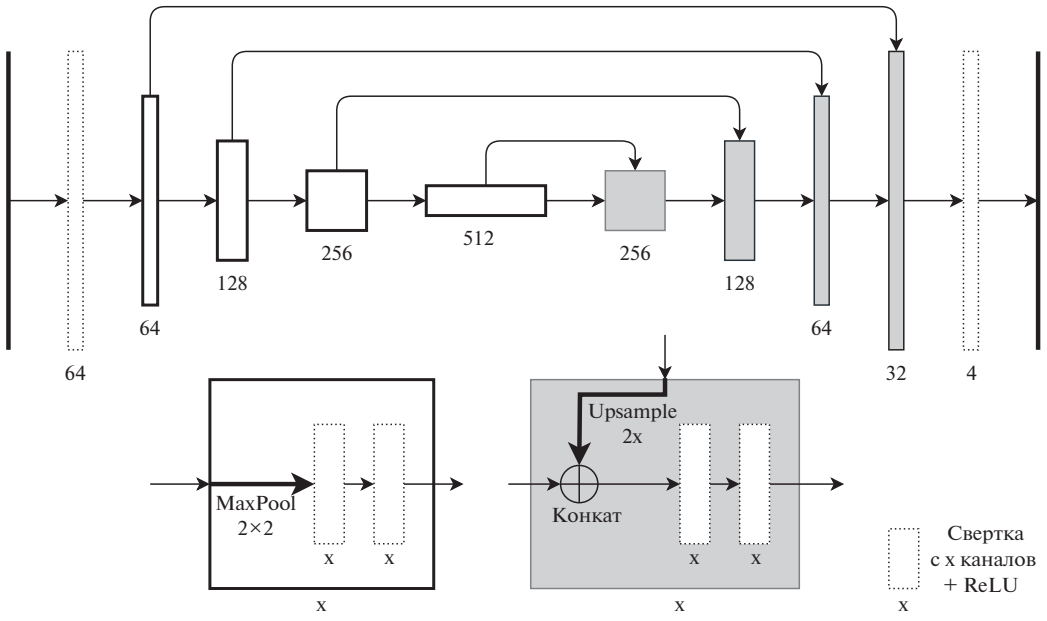


**Рис. 2.** Базовая архитектура сети.

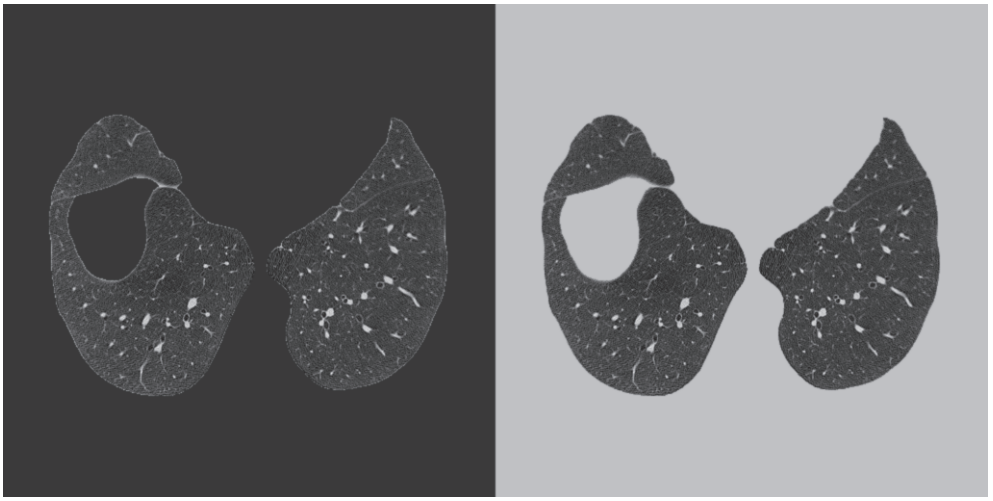

**Рис. 3.** Пример среза маскированного константой, соответствующей воздуху (слева) и ткани (справа).

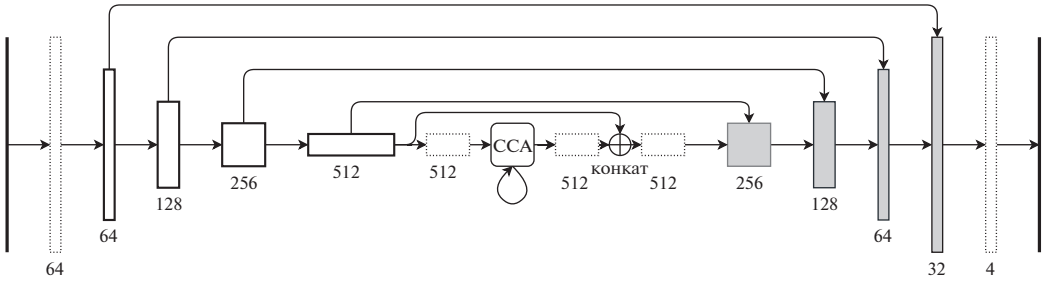


**Рис. 4.** Архитектура сети с модулем RCCA.

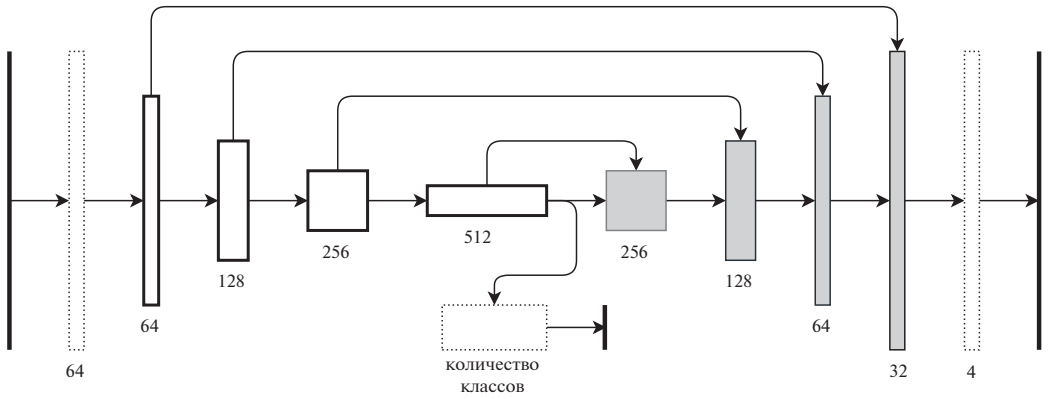


**Рис. 5.** Архитектура сети со стабилизирующей функцией потерь.

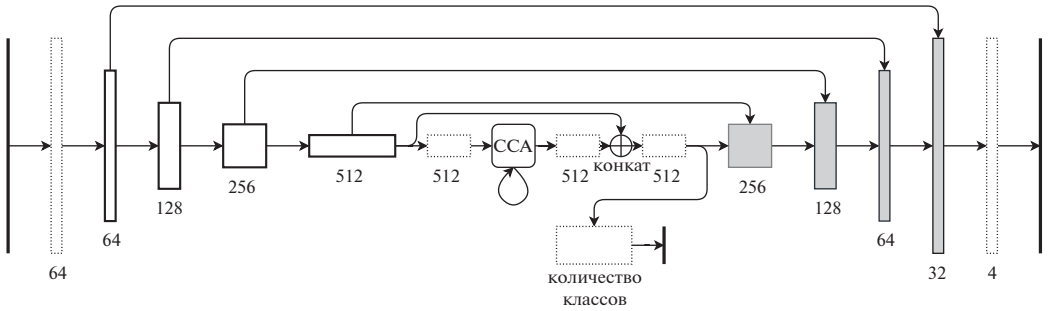


**Рис. 6.** Архитектура сети с модулем RCCA и стабилизирующей функцией потерь.

становится 0, что плохо отражает качество работы модели. Также небольшие ошибки в ответах будут ухудшать метрики на исследованиях с маленьким количеством истинно положительных пикселей гораздо сильнее, чем с большим.

Поэтому было принято решение считать общие метрики по 3 группам исследований: на исследованиях с отсутствующим классом считать среднее количество ложноположительных пикселей на исследование. На исследованиях, где пиксели класса есть, но их меньше 10000 на все исследование и на остальных считать mAP и IoU раздельно.

### *4.2. Метрики областей*

В результате диалога с радиологами оказалось, что даже при наличии улучшения модели по общим метрикам, врачи иногда не только не замечали этого улучшения, но еще и считали, что модель работает менее точно. Одной из причин этого является то, что модель находила участки легких и патологий вне легких, например, выделяла области кишечника. Для этого для каждого класса выделялись связанные компоненты и для каждой считалось, пересекается ли она с выделенными радиологами участками легких. Если не пересекается, то считаем, что компонента была определена вне легких. В качестве метрики будем использовать количество компонент вне легких разного размера: маленьких (до 100 пикселей), средних (от 100 до 500) и больших (более 500). Больше внимания стоит уделять именно большим компонентам, поскольку они заметны радиологам больше всего.

## 5. ПРЕДЛОЖЕННЫЙ МЕТОД

В данной работе решается задача уменьшения компонент, найденных вне легких, поэтому будут предложены способы устранения таких компонент. Базовым решением является сеть, подобная U-Net [7], архитектура представлена на рис. 2 обученная на классах “фон”, “легкое”, “патология”, “плевральный выпот” с функцией потерь отрицательная кросс-энтропия.

### *5.1. Предварительная сегментация легких*

Одним из подходов является предварительная сегментация участков легких, а затем поиск патологий в маскированных легких. Плюсом такого подхода будет то, что при корректной сегментации легких компоненты вне них не найдутся. Также сегментацию легких можно проводить на

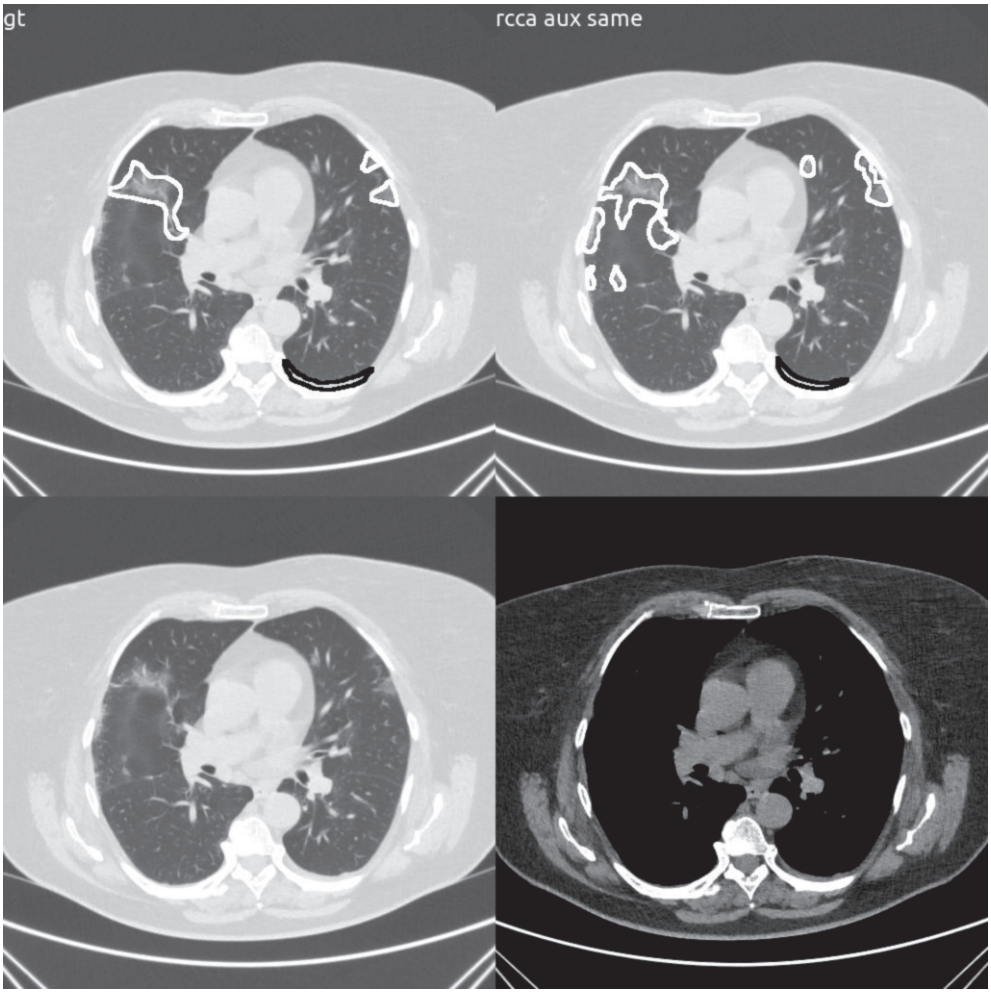


**Рис. 7.** Пример работы алгоритма на изображении с плевральным выпотом.

изображениях меньшего масштаба, что повысит точность сегментации, поскольку легкие занимают достаточно большую часть среза, в отличие от патологий. Минусами такого подхода является то, что в таком случае плевральный выпот будет маскирован, поскольку он не является частью легкого, а также то, что возрастут вычислительные и временные расходы на сегментацию легких.

Отдельным вопросом стоит то, какой константой следует заполнять маскированную область. С одной стороны ее можно заполнить значением, которое соответствует воздуху, а с другой стороны ткани, которая окружает легкие. Было опробовано оба варианта, пример можно видеть на рис. 3.

### *5.2. Введение модуля "RCCA"*

Этот модуль был описан в статье CCNet: Criss-Cross Attention for Semantic Segmentation [8]. Основная идея внедрения этого модуля заключается в том, чтобы в какой-то момент информация перераспределялась со всего изображения, чтобы сеть смогла более корректно отличать патологии и легкие от похожих на них структур по информации из контекста со всего изображения. В этой работе модуль помещен между энкодером и декодером сети U-Net, как показано на рис. 4.

### *5.3. Введение стабилизирующей функции потерь*

Также было предожено вставить дополнительную голову сети после энкодера или после модуля RCCA (при наличии) и вычислять грубую сегментацию в маленьком размере для того, чтобы сеть научилась не выделять большие компоненты в неправильных местах. Предложено предсказывать два варианта сегментаций и сравнить их между собой. Можно предсказывать ту же самую карту, что и после декодера, но в меньшем разрешении и использовать ту же функцию потерь, а можно предсказывать вероятность того, что пиксель принадлежит к интересующим нас классам и использовать бинарную кросс-энтропию.

**Таблица 1.** Сравнение базового (baseline) метода и методов с предварительной сегментацией для класса “патология”. segm air – сегментация с заполнением константой, соответствующей воздуху, segm body – сегментация с заполнением константой, соответствующей плотности тела

| | FP (0) | IoU (pat, s) | IoU (pat, l) | mAP (pat, s) | mAP (pat, l) |
|---|---|---|---|---|---|
| baseline | **575.6** | **0.413** | **0.603** | **0.656** | **0.856** |
| segm air | 641.5 | 0.402 | 0.596 | 0.626 | 0.840 |
| segm body | 783.6 | 0.374 | 0.584 | 0.604 | 0.833 |

**Таблица 2.** Сравнение модулей по количеству обнаруженных связанных компонент вне легких

| | Легкие | | | Патологии | | |
|---|---|---|---|---|---|---|
| | small comp | med comp | big comp | small comp | med comp | big comp |
| baseline | **4718** | 207 | 78 | 1051 | 393 | 52 |
| rcca | 5822 | 181 | 68 | 862 | 339 | 55 |
| aux fore | 6085 | **120** | **21** | 1144 | **237** | **20** |
| aux same | 6379 | 140 | 42 | 952 | 291 | 37 |
| rcca aux fore | 6356 | 152 | 30 | **835** | 354 | 47 |
| rcca aux same | 5788 | 128 | 31 | 736 | 330 | 41 |

## 6. РЕЗУЛЬТАТЫ

В таблицах ниже class_name, 0 означает исследования, на которых класс отсутствует, class_name, s – исследования, на которых в разметке содержится менее 10000 пикселей класса, class_name, l – исследования, на которых в разметке содержится более 10000 пикселей класса.

Так же приняты обозначения baseline – базовый метод, rcca в названии означает методы, использующие rcca, aux в названии означает наличие стабилизирующей функции потерь: aux fore для бинарной кросс-энтропии, различающей фон и интересующие классы, aux same для кросс-энтропии после софтмакс, для предсказывания аналогичных требуемым классам, но меньшего размера.

### *6.1. Исследование влияния предварительной сегментации легких*

Исследования проводились для моделей, которые предсказывали класс “патология” на предварительно маскированных легких.

Как видно из табл. 1, сегментация ухудшает ситуацию и на тех срезах, где патологий нет, поскольку начинает выдавать больше ложноположительных результатов, и на срезах, где патологии присутствовали.

Отдельно стоит отметить, что на результат может сильно влиять выбранная константа для заполнения маскированных регионов, поэтому стоит заострять на этом внимание, если используется подобная техника.

### *6.2. Исследование влияния модуля “RCCA” и стабилизирующей функции потерь*

Исследования проводились для моделей, которые предсказывали классы “легкое”, “патология”, “плевральный выпот”.

Для начала проанализируем то, как повлияло включение модуля на нахождение участков классов вне легких. Результаты представлены в табл. 2.

А также проанализируем влияние на общие метрики качества сегментации для классов “патология” (табл. 3) и “плевральный выпот” (табл. 4).

Сравнивая значения метрик, получаем, что применение стабилизирующих функций потерь уменьшает количество участков, найденных вне легких. При этом она чуть понижает значения метрик, которые были получены на изображениях с малым количеством патологий, зато увеличивает значения метрик, которые были получены на изображениях с большим количеством патологий. Также видно, что модуль RCCA сам по себе ухудшает значения метрик, но если поставить на его выходы стабилизирующую функцию потерь, то по общим метрикам он выигрывает на изображениях с большим количеством патологий и не сильно проигрывает на изображениях с малым количеством патологий, а также уменьшает количество ложных участков вне легких.

Примеры работы модели можно увидеть на рис. 7. В левой верхней строке слева разметка радиологов, справа результат работы модели, в нижней строке оригинальный срез с разными настройками просмотра.

**Таблица 3.** Сравнение модулей по общим метрикам для класса патологий

| | FP (pat, 0) | IoU (pat, s) | IoU (pat, l) | mAP (pat, s) | mAP (pat, l) |
|---|---|---|---|---|---|
| baseline | 575.6 | **0.413** | 0.603 | **0.656** | 0.856 |
| rcca | **525.9** | 0.412 | 0.615 | 0.627 | 0.856 |
| aux fore | 635.0 | 0.408 | 0.610 | 0.636 | 0.850 |
| aux same | 547.1 | **0.413** | 0.604 | 0.643 | 0.848 |
| rcca aux fore | 618.8 | 0.400 | 0.620 | 0.638 | 0.856 |
| rcca aux same | 613.9 | 0.409 | **0.627** | **0.656** | **0.861** |

**Таблица 4.** Сравнение модулей по общим метрикам для класса плеврального выпота

| | FP (pl eff, 0) | IoU (pl eff, s) | IoU (pl eff, l) | mAP (pl eff, s) | mAP (pl eff, l) |
|---|---|---|---|---|---|
| baseline | 961.6 | 0.383 | 0.632 | 0.590 | 0.803 |
| rcca | 407.9 | **0.320** | 0.636 | 0.556 | 0.807 |
| aux fore | 837.2 | 0.333 | 0.624 | 0.547 | 0.798 |
| aux same | 757.3 | 0.374 | 0.652 | 0.524 | **0.814** |
| rcca aux fore | **385.9** | 0.326 | 0.630 | 0.572 | 0.805 |
| rcca aux same | 698.7 | 0.369 | **0.660** | **0.596** | 0.812 |

## 7. ЗАКЛЮЧЕНИЕ

При оценке автоматической системы, которой будут пользоваться люди, необходимо пользоваться не только “классическими метриками” сегментации, но и метриками, которые будут оценивать применимость моделей для использования их людьми. Обычно при сравнении новых моделей, полученных в результате экспериментов, не получается модели, которая лучше остальных моделей во всем, поэтому стоит искать компромиссы.

## СПИСОК ЛИТЕРАТУРЫ


1. *Wang L., Lin Z. Q., Wong A.* Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest *x*-ray images // Scientific Reports. 2020. V. 10. № 1. P. 1–12.
2. *Li, L., Qin, L., Xu, Z. et al.* Artificial Intelligence Distinguishes COVID-19 from Community Acquired Pneumonia on Chest CT // Radiology. 2020.
3. *Shan F., Gao Y. et al.* Lung infection quantification of covid-19 in ct images with deep learning, arXiv preprint arXiv:2003.04655, 2020.
4. *Yazdekhasty P., Zindar A., Nabizadeh-ShahreBabak Z., Roshandel R., Khadivi P., Karimi N., Samavi S.* Bifurcated Autoencoder for Segmentation of COVID-19 Infected Regions in CT Images, arXiv preprint arXiv:2011.00631, 2020.
5. *Amyar A., Modzelewski R., Li H., Ruan S.* Multi-task deep learning based CT imaging analysis for COVID-19 pneumonia: Classification and segmentation // Computers in Biology and Medicine. 2020. V. 126. P. 104037.
6. Third Opinion Platform, Limited Liability Company, https://thirdopinion.ai/.
7. *Ronneberger O., Fischer P., Brox T.* U-net: Convolutional networks for biomedical image segmentation // International Conference on Medical image computing and computerassisted intervention. Springer, 2015. P. 234–241.
8. *Huang Z., Wang X., Huang L., Huang Ch., Wei Y., Liu W.* Ccnet: Criss-cross attention for semantic segmentation // Proceedings of the IEEE International Conference on Computer Vision. 2019. P. 603–612.