# ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

УДК 004.056.53 ISSN 1995-5499
DOI: https://doi.org/10.17308/sait.2020.1/2595
Поступила в редакцию 25.02.2020
Подписана в печать 15.03.2020

## АЛГОРИТМ КЛАССИФИКАЦИИ ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

**© 2020 А. В. Козачок, А. А. Спирин**✉

*Академия ФСО России*
*ул. Приборостроительная, 35, 302034 Орел, Российская Федерация*

**Аннотация.** В последнее время увеличилось количество утечек информации, произошедших по вине внутренних нарушителей, одной из возможных причин может являться не способность современных DLP систем противостоять утечкам информации в зашифрованном или сжатом виде. Был предложен алгоритм классификации последовательностей, сформированных алгоритмами шифрования, сжатия и генераторами псевдослучайных чисел. Для решения задачи классификации предла-гается использовать методы машинного обучения на основе алгоритма построения дерева решений. В качестве признакового пространства использовался массив ча-стот встречаемости двоичных подпоследовательностей длины N бит. При построе-нии признакового пространства не использовались заголовки файлов или какая-либо другая контекстная информация. Был обоснован выбор гиперпараметров классификатора. Представленный алгоритм показал точность классификации указанных в работе последовательностей 0.98. Представленный алгоритм может быть реализован в DLP системах для предотвращения передачи информации в зашифрованном или сжатом виде.

**Ключевые слова:** статистический анализ данных, машинное обучение, классификация бинарных последовательностей, DLP системы, защита информации от утечки.

### ВВЕДЕНИЕ

В последнее время, согласно отчетов информационно-аналитических агентств, возросло количество инцидентов, связанных с утечкой информации по вине внутренних нарушителей [1].

В работах [2, 3] отмечается, что причиной утечки конфиденциальных данных могут являться различные факторы: широкое распространение информационных технологий практически во все процессы обработки и передачи данных, внедрение удаленных рабочих мест, недостаточный уровень подготовки сотрудников в сфере информационной безопасности, несоблюдение комплекса организационных мер и др. Наибольшую угрозу, представляют внутренние нарушители, т. к. их действия не анализируются средствами защиты, направленными на отражение внешних атак. Внутренние нарушители отсекаются, в основном, DLP (data leakage prevention) системами.

В работах [4, 5] отмечается, что защита данных от внутренних нарушителей является сложной задачей, что подтверждается отсутствием у DLP систем механизмов анализа зашифрованных или сжатых данных, в случае отсутствия информации об алгоритме сжатия [6, 7].

В работе [8] авторы выделяют 2 группы методов, используемых в DLP системах: контентные и контекстные. Контентные методы

✉ Спирин Андрей Андреевич
e-mail: spirin_aa@bk.ru

используют для обнаружения конфиденциальных данных семантический анализ передаваемой информации, сигнатурный поиск, поиск цифровых слепков и регулярных выражений [9–12]. Контекстные методы используют метаданные [13]. В работе [14] авторы предлагают использовать поведенческие методы, осуществляющие формирование шаблонов стандартных действий пользователей или процессов при работе с данными, которые будут отличаться от действий нарушителей.

В работе [15] авторы рассматривают метод предотвращения утечек информации на основе контекстной целостности. В основе метода лежит идея легитимных информационных потоков.

В работах [16–20] рассмотрены методы идентификации криптоалгоритмов в различных режимах работы. Классификаторы, обученные на признаковых пространствах, сформированных в ходе выполнения подсчетов частот встречаемости различных подпоследовательностей символов, байт или бит являются одним из решений задачи идентификации криптоалгоритмов.

В работе [21] применяются сверточные нейронные сети GoogleNet, AlexNet для задачи бинарной классификации алгоритмов шифрования AES, DES в режиме простой замены и простой замены с зацеплением. Обе сети показали высокие результаты классификации с точностью более 0.9, однако сеть GoogleNet имеет более высокие значения точности на некоторых парах криптоалгоритмов.

Схожая задача классификации вредоносного трафика методами машинного обучения решалась в работах [22–26]. В работах [22–24] использовались методы, основанные на сверточных нейронных сетях, главное достоинство которых, по сравнению со стандартными алгоритмами машинного обучения, заключается в отсутствии необходимости осуществлять поиск и построение признакового пространства в явном виде. В работе [25] авторы предложили использовать комбинацию алгоритмов машинного обучения с учителем и без учителя для преодоления уязвимости нулевого дня, когда осуществляется ранее неизвестный тип атаки. В работе [26] приведен обзор методов машинного обучения, используемых для классификации трафика, описаны этапы обучения и построения классификаторов.

В работе [27] проводится сравнительный анализ методов машинного обучения на основе нейронных сетей для классификации зашифрованных и сжатых данных. Наибольшую точность в 66.9 % показала сверточная нейронная сеть, последовательная нейронная сеть показала точность в 54.1 %, метод k-ближайших соседей — 60 %. Данные результаты позволяют сделать вывод о необходимости исследования применимости других методов машинного обучения для решения задачи классификации зашифрованных и сжатых данных.

## 1. ПОСТАНОВКА ЗАДАЧИ ИССЛЕДОВАНИЯ

В общем виде задача классификации псевдослучайных последовательностей (ПСП) представлена в выражении 1 и формулируется следующим образом: необходимо исходное множество ПСП $X$ отобразить на множество классов $Y$ на основе классификатора, обученного на выбранном признаковом пространстве

$$F : X \in \{x_1, \ldots, x_j\} \to Y \in \{y_1, \ldots, y_i\}, \quad (1)$$

где $X$ — исходное множество бинарных ПСП, подлежащих классификации, $Y$ — множество классов, $F$ — функция отображения классификатора.

Множество классов $Y$ включает в себя:

1. Зашифрованные последовательности.
2. Сжатые последовательности.
3. Зашифрованные сжатые последовательности.
4. Последовательности, сформированные генераторами псевдослучайных чисел.

## 2. МЕТОДИКА ОЦЕНКИ КАЧЕСТВА КЛАССИФИКАТОРА

Для оценки качества классификатора используются следующие множества:

1. TP (true positive) — количество верно классифицированных ПСП, принадлежащих классу $y_i \in Y$.

2. TN (true negative) — количество ПСП, верно отнесенных не к классу $y_i \in Y$.

3. FP (false positive) — количество ПСП неверно отнесенных к классу $y_i \in Y$, т. е. количество ложных срабатываний (ошибка первого рода).

4. FN (false negative) — количество ПСП неверно не отнесенных к классу $y_i \in Y$, т. е. количество пропусков цели (ошибка второго рода).

Для оценки качества классификации была использована метрика доля правильных ответов, в общем виде определяемая выражением (2)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \qquad (2)$$

Для выборки, состоящей из $K$ классов ПСП, доля правильных ответов классификатора определяется выражением (3)

$$Accuracy_{total} = \frac{\sum_{i=1}^{K} Accuracy_{y_i}}{K}, \qquad (3)$$

где $Accuracy_{y_i}$ — доля правильных ответов для класса $y_i$.

С целью определения доли правильных ответов каждого класса строится матрица ошибок, представленная в табл. 1.

*Таблица 1. Матрица ошибок при классификации 4-х классов ПСП [Table 1. Confusion matrix for classification 4 class of PRS (PseudoRandomSequence)]*

| | | Истинный класс | | | |
|---|---|---|---|---|---|
| Предсказанный класс | K | 1 | 2 | 3 | 4 |
| | 1 | $T_1$ | $F_{12}$ | $F_{13}$ | $F_{14}$ |
| | 2 | $F_{21}$ | $T_2$ | $F_{23}$ | $F_{24}$ |
| | 3 | $F_{31}$ | $F_{32}$ | $T_3$ | $F_{33}$ |
| | 4 | $F_{41}$ | $F_{42}$ | $F_{43}$ | $T_4$ |

При проведении многоклассовой классификации подсчет множеств осуществляется на основе матрицы ошибок по формулам (4):

$$\begin{cases} TP_{y_i} = T_{y_i} \\ TN_{y_i} = \sum_{c=1}^{K} T_c - TP_{y_i} \\ FP_{y_i} = \sum_{c=1}^{K} F_{y_i,c} \\ FN_{y_i} = \sum_{c=1}^{K} F_{c,y_i} \end{cases}, \qquad (4)$$

где $y_i$ — истинный класс ПСП, $c$ — предсказанный классификатором класс.

Значение доли правильных ответов для выбора классификатора должно удовлетворять условию, представленному в выражении (5):

$$Accuracy_{total} \to 1. \qquad (5)$$

Для классификации ПСП предлагается использовать алгоритм, основанный на подсчете количества двоичных подпоследовательностей длины $N-1$ бит в исследуемых ПСП. В работах [28, 29] отмечается, что для Например, для последовательности $s = 1011010001$ частота вхождения подпоследовательностей длины $N = 3$ бит представлена в табл. 2.

восстановления распределения бинарных последовательностей достаточно анализировать половину всех возможных подпоследовательностей. Таким образом размерность признакового пространства для подпоследовательностей длины $N$ бит определяется выражением (6):

$$|S| = 2^{N-1}. \qquad (6)$$

*Таблица 2. Подсчет частот подпоследовательностей [Table 2. Subsequences frequencies counting]*

| Подпоследовательность | Кол-во | Частота |
|---|---|---|
| 000 | 1 | 0,125 |
| 001 | 1 | 0,125 |
| 010 | 1 | 0,125 |
| 011 | 1 | 0,125 |

## 3. АЛГОРИТМ ПОСТРОЕНИЯ ПРИЗНАКОВОГО ПРОСТРАНСТВА

Алгоритм классификации ПСП состоит из 3-х этапов: формирование признакового пространства, построение на его основе классификатора, применение полученного классификатора к исследуемым данным.

### 3.1. Алгоритм построения признакового пространства

Исходными данными для алгоритма построения признакового пространства являются: размеченное на классы $Y$ множество

ПСП $P$ мощностью $Q$, множество двоичных подпоследовательностей $S$ длины $N$ бит мощностью $2^{N-1}$. Множество $S$ формируется путем построения всех возможных двоичных подпоследовательностей заданной длины $N$ бит.

Алгоритм построения признакового пространства представлен на рис. 1.

**Data:** P: |P|=Q, S: $|S| = 2^{N-1}$
**Result:** $F_{Q,E}$
1 $F_{Q,E} \leftarrow <>$
2 **for** $p \in P$ **do**
3 $\quad M_p \leftarrow \mathbf{Len}(p)$
4 $\quad$ **for** $s \in S$ **do**
5 $\quad\quad n_s \leftarrow \mathbf{Count}(p,s)$
6 $\quad\quad f_{p,s} \leftarrow \frac{n_s}{M_p - N_s + 1}$
7 $\quad\quad F_{Q,E} \leftarrow F_{Q,E} \cup < f_{p,s}, y_i >$
8 return $F_{Q,E}$

*Рис. 1. Алгоритм построения признакового пространства*
*[Fig. 1. Features space building algorithm]*

**Шаг 1.**

Инициализировать пустой кортеж частот подпоследовательностей $F_{Q,E}$.

**Шаг 2.**

Для каждой ПСП $p$ из множества $P$ выполнить:

1. Определить длину подпоследовательности $p$ и присвоить ее значение переменной $M_p$.

Для каждой подпоследовательности $s$ из множества $S$ выполнить:

2. Переменной $n_s$ присвоить значение функции $Count(p,s)$. Функция выполняет подсчет количества вхождений подпоследовательности $s$ в ПСП $p$ без перекрытия.

3. Переменной $f_{p,s}$ присвоить значение выражения $\frac{n_s}{(M_p - N_s + 1)}$, где $n_s$ — количество вхождений подпоследовательности $s$ в ПСП $p$ без перекрытия, $M_p$ — длина ПСП $p$ в битах, $N_s$ — длина подпоследовательности $s$ в битах.

4. Записать в кортеж $F_{Q,E}$ значение частоты вхождения подпоследовательности $s$ в ПСП $p$ $f_{p,s}$ и связанный с подпоследовательностью $p$ класс ПСП $y_i$.

**Шаг 3.**

Возвратить кортеж $F_{Q,E}$.

Полученный кортеж значений частот встречаемости подпоследовательностей длины $N$ бит является признаковым пространством для дальнейшего обучения и построения классификатора.

### 3.2. Алгоритм построения классификатора

Исходными данными для построения классификатора являются:

1. Выборка частот подпоследовательностей длины $N$ бит в исходном множестве ПСП.

2. Максимальная глубина дерева $D$.

Алгоритм построения классификатора представлен на рис. 2.

**Шаг 1.**

1. Текущий уровень разбиения дерева $d$ принять равным 1.

2. Принять массив множеств частот подпоследовательностей длины $N$ бит $Z$ равным исходной обучающей выборке $F_{Q,E}(|Z| = 1)$.

3. Проинициализировать пустыми значениями кортеж классификатора $K$ и кортеж признаков $V$.

4. Задать начальное значение нумератора узлов в классификаторе $counter = 0$.

**Шаг 2.**

Для каждого целого шага разбиения дерева $d$ в интервале от 0 до заданного максимального значения глубины дерева $D$, для каждого элемента массива множеств $Z$ выполнить:

1. Инициализировать индекс Джини $G_{index} = 0$.

2. Инициализировать кортеж узлов *nodes* пустым значением.

3. Для каждой подпоследовательности $e \in E$ выполнить:

3.1. Отсортировать по возрастанию значения частот вхождения подпоследовательности $e$ во множество ПСП $\{Q\}$.

3.2. Для каждой ПСП $q$ в интервале $[1; Q-1]$ выполнить:

3.2.1. Инициализировать кортеж потомков левой части разбиения $z_{g\_left}$ пустым значением.

3.2.2. Инициализировать кортеж потомков правой части разбиения $z_{g\_right}$ пустым значением.

3.2.3. Определить пороговое значение частоты вхождения подпоследовательности $e \in E$ в ПСП $q$ по формуле $T_q = \frac{f_{q,e} + f_{q+1,e}}{2}$.

3.2.4. Для каждой ПСП а в интервале $[1;Q]$ выполнить:

Если значение частоты вхождения подпоследовательности $e \in E$ в ПСП а меньше порогового значения $T_q$, то дополнить кортеж $z_{g\_left}$ множеством $\{f_{a,e}, y_a\}$, иначе дополнить кортеж $z_{g\_right}$ множеством.

3.2.5. Определить множество левых $L = \{z_{g\_left}\}$ и правых $R = \{z_{g\_right}\}$ потомков.

3.2.6. Рассчитать индекс Джини для текущего разбиения множества $z$ по формуле: $G_{current} = \left( \frac{1}{L} * \sum_{y_q=1}^{i} P^2(y_q) + \frac{1}{R} * \sum_{y_q=1}^{i} P^2(y_q) \right)$.

3.2.7. Если полученное значение индекса Джини больше установленного значения индекса Джини:

3.2.8. Присвоить значение $G_{index} = G_{current}$.

3.2.9. Определить метку класса множества $z$ как метку класса $y$, которая содержится во множестве $z$ в максимальном количестве.

3.2.10. Включить $< G_{current}, e, T_q, class, \{L\}, \{R\}, TN = False, counter >$ в кортеж $nodes$.

3.2.11. Определить узел разбиения по максимальному значению индекса Джини $host = node \in nodes : \forall n \in nodes,\ node[1] \geq n[1]$.

3.2.12. Добавить признак $e$ во множество $V$.

3.2.13. Присвоить узлу разбиения порядковый номер.

3.2.14. Удалить из множества $Z$ подмножество $z$.

3.2.15. Определить множества $L$, $R$ как подмножества $L$ и $R$ кортежа $host$.

3.2.16. Если в множестве $L$ содержится более одного класса меток классов ПСП: включить множество $L$ в множество $Z$.

3.2.17. Иначе:

3.2.18. Добавить флаг терминального узла в множество $L$.

3.2.19. Присвоить множеству $L$ значение множества $host$ без множества $R$.

3.2.20. Добавить в классификатор $K$ с помощью функции $AddGraph$ следующее ребро графа в виде текущего узла $L$ и родительского узла $host$.

3.2.21. Если в множестве $R$ содержится более одного класса меток классов ПСП: Включить множество $R$ в множество $Z$.

3.2.22. Иначе:

3.2.23. Добавить флаг терминального узла в множество $R$.

3.2.24. Присвоить множеству $R$ значение множества $host$ без множества $L$.

3.2.25. Добавить в классификатор $K$ с помощью функции $AddGraph$ следующее ребро графа в виде текущего узла $R$ и родительского узла $host$. Если значение нумератора узлов не равно 0, выполнить: добавить в классификатор $K$ с помощью функции $AddGraph$ следующее ребро графа в виде текущего узла $host$ и родительского узла, являющегося вторым элементом с конца множества $host$.

3.2.26. Иначе добавить в классификатор $K$ с помощью функции $AddGraph$ корневой узел $host$.

3.2.27. Увеличить значение нумератора узлов классификатора на 1.

3.2.28. Увеличить значение счетчика узлов классификатора на 1.

**Шаг 3.**

Возвратить кортеж $K$ в виде графа и множество признаков $V$.

### 3.3. Классификация ПСП

Исходными данными для выполнения классификации ПСП $p$ являются:

1. ПСП $p$.
2. Классификатор $K$, множество признаков $V$.

Алгоритм классификации ПСП представлен на рис. 3.

**Шаг 1.**

1. Инициализировать кортеж $F_{Q,V}$ пустым значением.

2. Инициализировать кортеж состояния $State$ пустым значением.

3. Вычислить длину последовательности $p$ $M_p$ в битах.

**Шаг 2.**

Для всех признаков $v$ из кортежа $V$ выполнить:

1. Вычислить длину подпоследовательности $v$ и записать полученное значение в переменную $N_v$.

2. Вычислить количество вхождений подпоследовательности $v$ в ПСП $p$ и записать полученное значение в переменную $n_v$.

3. Вычислить частоту вхождения подпоследовательности $v$ в ПСП $p$ по формуле $\frac{n_v}{(M_p - N_v + 1)}$.

4. Добавить значение частоты подпоследовательности $v$ в ПСП $p$ в кортеж $F_{Q,V}$.

**Data:** $F_{Q,E} =< f_{q,e}, y_q \in Y >$, D=const
**Result:** K

$d \leftarrow 1$
$Z \leftarrow F_{Q,E}, |Z| = 1$
$K \leftarrow <>$
$V \leftarrow <>$
$counter \leftarrow 0$
**for** $d \in 1..D$ **do**
  **for** $z \in Z$ **do**
    $G_{index} \leftarrow 0$
    $nodes \leftarrow <>$
    **for** $e \in E$ **do**
      $z \leftarrow$ **SortAcsending**$(z,e)$
      **for** $q \in Q-1$ **do**
        $z_{left} \leftarrow <>$
        $z_{right} \leftarrow <>$
        $T_q \leftarrow \frac{f_{q,e} + f_{q+1,e}}{2}$
        **for** $a \in Q$ **do**
          **if** $f_{a,e} \leq T_q$ **then**
            $z_{left} \cup \{f_{a,e}, y_a\}$
          **else**
            $z_{right} \cup \{f_{a,e}, y_a\}$
        $L \leftarrow |z_{left}|$
        $R \leftarrow |z_{right}|$
        $G_{current} \leftarrow \left(\frac{1}{L} * \sum_{y_q=1}^{i} P^2(y_q) + \frac{1}{R} * \sum_{y_q=1}^{i} P^2(y_q)\right)$
        **if** $G_{current} > G_{index}$ **then**
          $G_{index} \leftarrow G_{current}$
          $class \leftarrow y_q : max\ |y_q| \in z$
          $nodes \leftarrow nodes \cup < G_{current}, e, T_q, class, \{L\}, \{R\}, TN \leftarrow False, counter >$
    $host \leftarrow node \in nodes : \forall n \in nodes, node[1] \geq n[1]$
    $V \leftarrow V \cup host[2]$
    $host[8] \leftarrow counter$
    $Z \leftarrow Z \setminus z$
    $L \leftarrow host[5]$
    $R \leftarrow host[6]$
    **if** $|Y_q| \in L \neq 1$ **then**
      $Z \leftarrow Z \cup L$
    **else**
      $host[7] \leftarrow True$
      **AddGraph**$(K, \{L, host\})$
    **if** $|Y_q| \in R \neq 1$ **then**
      $Z \leftarrow Z \cup R$
    **else**
      $host[7] \leftarrow True$
      **AddGraph**$(K, \{R, host\})$
    **if** $counter \neq 0$ **then**
      **AddGraph**$(K, \{host, host[-1]\})$
    **else**
      **AddGraph**$(K, \{host\})$
  $d \leftarrow d + 1$
  $counter \leftarrow counter + 1$
return {K,V}

*Рис. 2. Алгоритм построения классификатора*
*[Fig. 2. Classificator building algorithm]*

## 4. ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ

Для осуществления эксперимента была сформирована выборка ПСП, состоящая из 9000 файлов 4-х классов, полученных в результате преобразований файлов, содержащих осмысленный текст на русском языке:

1. Зашифрованные алгоритмами AES, 3DES, RC4, Camellia в режиме гаммирования с обратной связью [30] – 4000 файлов.

2. Архивы RAR, ZIP [31] — 2000 файлов.

3. Зашифрованные архивы RAR, ZIP [31] — 2000 файлов.

4. Сформированные утилитой urandom операционной системы семейства Linux [32] — 1000 файлов.

Эксперимент проводился в программной среде Anaconda [33].

**Data:** ПСП p, классификатор <K>, <V>
**Result:** Класс y ПСП p

$F_{Q,V} \leftarrow <>$
$State \leftarrow <>$
$M_p \leftarrow$ **Len**$(p)$
**for** $v \in V$ **do**
  $N_v \leftarrow$ **Len**$(v)$
  $n_v \leftarrow$ **Count**$(p,v)$
  $f_{p,v} = \frac{n_v}{M_p - N_v + 1}$
  $F_{Q,V} = F_{Q,V} \cup f_{p,v}$
$State \leftarrow$ **Next**$(k)$
**while** $State[7] \neq True$ **do**
  **if** $f_{p,State[2]} \geq State[3]$ **then**
    $State \leftarrow$ **NextRight**$(State)$
  **else**
    $State \leftarrow$ **NextLeft**$(State)$
$y_p \leftarrow State[4]$
return $y_p$

*Рис. 3. Алгоритм классификации ПСП*
*[Fig. 3. PRS classification algorithm]*

Поскольку полученные значения частоты встречаемости подпоследовательностей длины N бит являются достаточно малыми величинами $(\sim 10^{-5}..10^{-6})$, то был осуществлен переход к логарифмическому масштабу значений для повышения точности классификации (логарифмические признаки).

Для построения классификаторов и проведения их оценки были применены алгоритмы машинного обучения [34]: классификатор на основе дерева решений (КДР), классификатор на основе дерева решений на логарифмических признаках (КДРЛ), классификатор на основе случайного леса (КСЛ), классификатор на основе случайного леса на логарифмических признаках (КСЛЛ). Полученные значения точности классификации ПСП от

длины подпоследовательности N представлены на рис. 4.

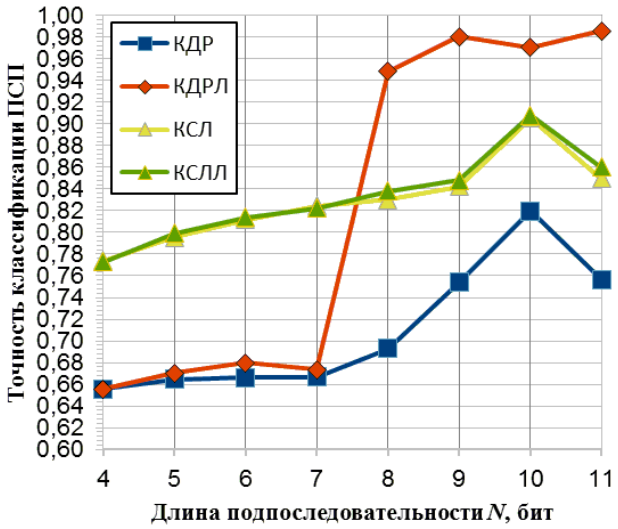


*Рис. 4. Точность классификации ПСП*
*[Fig. 4. Accuracy for classification PSR]*

Полученные результаты свидетельствуют о возможности классификации ПСП, сформированных алгоритмами шифрования, сжатия и генераторами псевдослучайных чисел предложенным алгоритмом с точностью более 0.95 при длине подпоследовательности 9 бит.

Особое влияние на точность классификации оказал переход к логарифмическому масштабу значений частот встречаемости подпоследовательностей в ПСП и позволил повысить точность классификатора на основе алгоритма построения дерева решений до 0.98.

## ЗАКЛЮЧЕНИЕ

Поскольку современные DLP системы допускают возможность передачи конфиденциальной информации в зашифрованном или сжатом виде был предложен алгоритм классификации последовательностей, сформированных криптоалгоритмами, алгоритмами сжатия данных и генераторами псевдослучайных чисел.

В ходе проведения экспериментов использовалось 2 алгоритма построения классификаторов: алгоритм построения дерева решений и алгоритм построения случайного леса. Алгоритм построения дерева решений показал более высокую точность классификации ПСП. Для повышения точности классификатора значения частот встречаемости подпоследовательностей были переведены в логарифмический масштаб, что позволило достичь точности классификации ПСП в 0.98.

## КОНФЛИКТ ИНТЕРЕСОВ

Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

## СПИСОК ЛИТЕРАТУРЫ

1. Data Breach Report: A Study on Global Data Leaks in H1 2018 / InfoWatch. – Режим доступа: https://www.infowatch.ru/analytics/reports. – (Дата обращения 14.01.2020)

2. *Babu, B. M.* Prevention of Insider Attacks by Integrating Behavior Analysis with Risk based Access Control Model to Protect Cloud / B. M. Babu, M. S. Bhanu // Procedia Computer Science. – 2015. – V. 54. – P. 157–166. DOI: 10.1016/j.procs.2015.06.018

3. *Kolevski, D.* Cloud computing data breaches a socio-technical review of literature / D. Kolevski, K. Michael // 2015 International Conference on Green Computing and Internet of Things (ICGCIoT). – Greater Noida, India, 2015. – P. 1486–1495. DOI: 10.1109/ICGCIoT.2015.7380702

4. *Alneyadi, S.* Detecting Data Semantic: A Data Leakage Prevention Approach / S. Alneyadi, E. Sithirasenan, V. Muthukkumarasamy // 2015 IEEE Trustcom/BigDataSE/ISPA. – Helsinki, Finland, 2015. – V. 1. – P. 910–917. DOI: 10.1109/Trustcom.2015.464

5. *Alneyadi, S.* Discovery of potential data leaks in email communications / S. Alneyadi, E. Sithirasenan, V. Muthukkumarasamy // 2016 10th International Conference on Signal Processing and Communication Systems (ICSPCS). – Gold Coast, Australia, 2016. – P. 1–10. DOI: 10.1109/ICSPCS.2016.7843323

6. *Huang, X.* A novel mechanism for fast detection of transformed data leakage / X. Huang, Y. Lu, D. Li, M. Ma // IEEE Access. – 2018. –

V. 6. – P. 35926–35936. DOI: 10.1109/ACCESS.2018.2851228

7. *Kaur, K.* A Comparative Evaluation of Data Leakage/Loss prevention Systems (DLPS) / K. Kaur, I. Gupta, A. K. Singh // In Proc. 4th Int. Conf. Computer Science & Information Technology (CS & IT-CSCP). – 2017. – P. 87–95. DOI: 10.5121/csit.2017.71008

8. *Cheng, L.* Enterprise data breach: causes, challenges, prevention, and future directions / L. Cheng, F. Liu, D. Yao // Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. – 2017. – V. 7, № 5. – P. 1211 DOI: 10.1002/widm.1211

9. *Shu, X.* Privacy-Preserving Detection of Sensitive Data Exposure / X. Shu, D. Yao, E. Bertino // IEEE Transactions on Information Forensics and Security. – 2015. – V. 10, No 5. – P. 1092–1103. DOI: 10.1109/TIFS.2015.2398363

10. *Liu, F.* Privacy-preserving scanning of big content for sensitive data exposure with MapReduce / F. Liu, X. Shu, D. Yao, A. R. Butt // Proceedings of the 5th ACM Conference on Data and Application Security and Privacy. – 2015. – P. 195–206. DOI: 10.1145/2699026.2699106

11. *Shu, X.* Rapid and parallel content screening for detecting transformed data exposure / X. Shu, J. Zhang, D. Yao, W. Feng // Proceedings of the Third International Workshop on Security and Privacy in Big Data. – 2015. – P. 191–196. DOI: 10.1109/INFCOMW.2015.7179383

12. *Shu, X.* Fast Detection of Transformed Data Leaks / X. Shu [and others] // IEEE Transactions on Information Forensics and Security. – 2016. – V. 11, No 3. – P. 528–542. DOI: 10.1109/TIFS.2015.2503271

13. *Yu, X.* A data leakage prevention method based on the reduction of confidential and context terms for smart mobile devices / X. Yu [and others] // Wireless Communications and Mobile Computing. – 2018. DOI: 10.1155/2018/5823439

14. *Shu, X.* Privacy-Preserving Detection of Sensitive Data Exposure / X. Shu, D. Yao, E. Bertino // IEEE Transactions on Information Forensics and Security. – 2015. – V. 10, No 5. – P. 1092–1103. DOI: 10.1109/TIFS.2015.2398363

15. *Shvartzshnaider, Y.* VACCINE: Using Contextual Integrity For Data Leakage Detection / Y. Shvartzshnaider [and others] // The World Wide Web Conference. – 2019. – P. 1702–1712. DOI: 10.1145/3308558.3313655

16. Kavitha, T. Classification of encryption algorithms based on ciphertext using pattern recognition techniques / T. Kavitha [and others] // International conference on Computer Networks, Big data and IoT. – 2018. – P. 540–545. DOI: 10.1007/978-3-030-24643-3_64

17. *Tan, C.* An approach to identifying cryptographic algorithm from ciphertext / C. Tan, Q. Ji // 8th IEEE International Conference on Communication Software and Networks. – 2016. – P. 19–23. DOI: 10.1109/ICCSN.2016.7586649

18. *Tan, C.* A Novel Identification Approach to Encryption Mode of Block Cipher / C. Tan, Y. Li, S. Yao // 4th International Conference on Sensors, Mechatronics and Automation. – Zhuhai, China, 2016. DOI: 10.2991/icsma-16.2016.101

19. *Tan, C.* Identification of Block Ciphers under CBC Mode / C. Tan, X. Deng, L. Zhang // Procedia Computer Science. – 2018. – Vol. 131. – P. 65–71. DOI: 10.1016/j.procs.2018.04.186

20. *Ray, P. K.* Classification of Encryption Algorithms using Fisher's Discriminant Analysis / P. K. Ray [and others] //Defence Science Journal. – 2017. – V. 67, No 1. – P. 59-65. DOI : 10.14429/dsj.67.9153

21. *Pan, J.* Encryption scheme classification: a deep learning approach / J. Pan // International Journal of Electronic Security and Digital Forensics. – 2017. – V. 9, No 4. – P. 381–395. DOI: 10.1504/IJESDF.2017.087397

22. *Wang, W.* Malware traffic classification using convolutional neural network for representation learning / W. Wang [and others] // International Conference on Information Networking (ICOIN). – 2017. – P. 712–717. DOI: 10.1109/ICOIN.2017.7899588

23. *Wang, W.* End-to-end encrypted traffic classification with one-dimensional convolution neural networks / W. Wang [and others] // IEEE International Conference on Intelligence and Security Informatics (ISI). – 2017. – P. 43–48. DOI: 10.1109/ISI.2017.8004872

24. *Lotfollahi, M.* Deep packet: A novel approach for encrypted traffic classification using deep learning / M. Lotfollahi [and others] // Soft Computing. – 2017. – P. 1–14.

25. *Zhang, J.* Robust network traffic classification / J. Zhang [and others] // IEEE/ACM Transactions on Networking. – 2015. – V. 23, No 4. – P. 1257–1270. DOI: 10.1109/TNET.2014.2320577

26. *Pacheco, F.* Towards the deployment of machine learning solutions in network traffic classification: a systematic survey / F. Pacheco [and others] //IEEE Communications Surveys & Tutorials. – 2018. – V. 21. – No 2. – P. 1988–2014. DOI: 10.1109/COMST.2018.2883147

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

28. *Конышев, М. Ю.* Формирование распределений вероятностей двоичных векторов источника ошибок марковского дискретного канала связи с памятью с применением метода «группирования вероятностей» векторов ошибок / М. Ю. Конышев [и др.] // Промышленные АСУ и контроллеры. – 2018. – № 3. – С. 42.

29. *Конышев, М. Ю.* Алгоритм сжатия ряда распределения двоичных многомерных случайных величин / М. Ю. Конышев [и др.] // Промышленные АСУ и контроллеры. – 2016. – №. 8. – С. 47–50.

30. Toolkit for the transport layer security and secure sockets layer protocols. – Режим доступа: http://openssl.org. – (Дата обращения: 14.01.2020).

31. Archive manager WinRAR. – Режим доступа: http://rarlab.com. – (Дата обращения: 14.01.2020).

32. Linux programmer's manual. – Режим доступа: http://man7.org/linux/man-pages/man4/random.4.html. – (Дата обращения: 14.01.2020).

33. Программная среда Anaconda. – Режим доступа: https://www.anaconda.com/distribution/. – (Дата обращения: 14.01.2020).

34. Breiman, L. Classification and regression trees / L. Breiman. – Routledge, 2017. – 358 с.

**Козачок Александр Васильевич** — д-р техн. наук, сотрудник, Академия ФСО России.
E-mail: a.kozachok@academ.msk.rsnet.ru
https://orcid.org/0000-0002-6501-2008

**Спирин Андрей Андреевич** — сотрудник, Академия ФСО России.
E-mail: spirin_aa@bk.ru
https://orcid.org/0000-0002-7231-5728

DOI: https://doi.org/10.17308/sait.2020.1/2595 ISSN 1995-5499
Received 25.02.2020
Accepted 15.03.2020

# ALGORITHM FOR CLASSIFYING PSEUDO-RANDOM SEQUENCESALGORITHM FOR THE CLASSIFICATION OF PSEUDORANDOM SEQUENCES

**© 2020 A. V. Kozachok, A. A. Spirin**✉

*Russian Federation Security Guard Service Federal Academy*
*35 Priborostroitelnaya Str., 302034 Orel, Russian Federation*

**Abstract.** The number of information leaks caused by internal violators has increased recently. One of the causes may be the inability of modern DLP systems to prevent information leaks in encrypted or compressed form. The article suggests an algorithm for the classification of sequences generated by encryption and compression algorithms and pseudorandom number generators. To solve the classification problem, we suggest using machine learning methods based on a decision tree algo-rithm. An array of frequencies of binary subsequences of N bit length was used as a feature space. File headers or any other contextual information were not used when constructing the feature space. The choice of hyperparameters of the classifier was substantiated. The suggested algorithm showed the accuracy of classification of the described sequences to be equal to 0.98. The suggested algo-rithm can be implemented in DLP systems to prevent the transmission of information in encrypted or compressed form.
**Keywords:** statistical analysis of data, machine learning, classification of binary sequences, DLP systems, data leak prevention.

## CONFLICT OF INTEREST

The authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.

## REFERENCES

1. Data Breach Report: A Study on Global Data Leaks in H1 2018 / InfoWatch. Access mode: https://www.infowatch.ru/analytics/reports. (accessed 14.01.2020).

2. *Babu B. M., Bhanu M. S.* Prevention of Insider Attacks by Integrating Behavior Analysis with Risk based Access Control Model to Protect Cloud. Procedia Computer Science. 2015. V. 54. P. 157–166. DOI: 10.1016/j.procs.2015.06.018

3. *Kolevski D., Michael K.* Cloud computing data breaches a socio-technical review of literature. 2015 International Conference on Green Computing and Internet of Things (ICGCIoT). Greater Noida, India, 2015. P. 1486–1495. DOI: 10.1109/ICGCIoT.2015.7380702

4. *Alneyadi S., Sithirasenan E., Muthukkumarasamy V.* Detecting Data Semantic: A Data Leakage Prevention Approach. 2015 IEEE Trustcom/BigDataSE/ISPA. Helsinki, Finland, 2015. V. 1. P. 910–917. DOI: 10.1109/Trustcom.2015.464

5. *Alneyadi S., Sithirasenan E., Muthukkumarasamy V.* Discovery of potential data leaks in email communications. 2016 10th International Conference on Signal Processing and Communication Systems (ICSPCS). Gold Coast, Australia, 2016. P. 1–10. DOI: 10.1109/ICSPCS.2016.7843323

6. *Huang X., Lu Y., Li D., Ma M.* A novel mechanism for fast detection of transformed data leakage. IEEE Access. 2018. V. 6. P. 35926–35936. DOI: 10.1109/ACCESS.2018.2851228

7. *Kaur K., Gupta I., Singh A. K.* A Comparative Evaluation of Data Leakage/Loss prevention Systems (DLPS). In Proc. 4th Int. Conf. Computer Science & Information Technology (CS & IT-CSCP). 2017. P. 87–95. DOI: 10.5121/csit.2017.71008

✉ Spirin Andrey A.
e-mail: spirin_aa@bk.ru

8. *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, No 5. P. 1211. DOI: 10.1002/widm.1211

9. *Shu X., Yao D., Bertino E.* Privacy-Preserving Detection of Sensitive Data Exposure. IEEE Transactions on Information Forensics and Security. 2015. V. 10, No 5. P. 1092–1103. DOI: 10.1109/TIFS.2015.2398363

10. *Liu F., Shu X., Yao D., Butt A. R.* Privacy-preserving scanning of big content for sensitive data exposure with MapReduce. Proceedings of the 5th ACM Conference on Data and Application Security and Privacy. 2015. P. 195–206. DOI: 10.1145/2699026.2699106

11. *Shu X., Zhang J., Yao D., Feng W.* Rapid and parallel content screening for detecting transformed data exposure. Proceedings of the Third International Workshop on Security and Privacy in Big Data. 2015. P. 191–196. DOI: 10.1109/INFCOMW.2015.7179383

12. *Shu X. [et al]* Fast Detection of Transformed Data Leaks. IEEE Transactions on Information Forensics and Security. – 2016. V. 11, No 3. P. 528–542. DOI: 10.1109/TIFS.2015.2503271

13. *Yu X. [et al]* A data leakage prevention method based on the reduction of confidential and context terms for smart mobile devices. Wireless Communications and Mobile Computing. 2018. DOI: 10.1155/2018/5823439

14. *Shu X., Yao D., Bertino E.* Privacy-Preserving Detection of Sensitive Data Exposure. IEEE Transactions on Information Forensics and Security. 2015. V. 10, No 5. P. 1092–1103. DOI: 10.1109/TIFS.2015.2398363

15. *Shvartzshnaider Y. [et al]* VACCINE: Using Contextual Integrity For Data Leakage Detection. The World Wide Web Conference. 2019. P. 1702–1712. DOI: 10.1145/3308558.3313655

16. *Kavitha T. [et al]* Classification of encryption algorithms based on ciphertext using pattern recognition techniques. International conference on Computer Networks, Big data and IoT. 2018. P. 540-545. DOI: 10.1007/978-3-030-24643-3_64

17. *Tan C., Ji Q.* An approach to identifying cryptographic algorithm from ciphertext. 8th IEEE International Conference on Communication Software and Networks. 2016. P. 19–23. DOI: 10.1109/ICCSN.2016.7586649

18. *Tan C., Li Y., Yao S.* A Novel Identification Approach to Encryption Mode of Block Cipher. 4th International Conference on Sensors, Mechatronics and Automation. Zhuhai, China, 2016. DOI: 10.2991/icsma-16.2016.101

19. *Tan C., Deng X., Zhang L.* Identification of Block Ciphers under CBC Mode. Procedia Computer Science. 2018. Vol. 131. P. 65–71. DOI: 10.1016/j.procs.2018.04.186

20. *Ray P. K. [et al]* Classification of Encryption Algorithms using Fisher's Discriminant Analysis. Defence Science Journal. 2017. V. 67, No 1. P. 59–65. DOI : 10.14429/dsj.67.9153

21. *Pan J.* Encryption scheme classification: a deep learning approach. International Journal of Electronic Security and Digital Forensics. 2017. V. 9, No 4. P. 381–395. DOI: 10.1504/IJESDF.2017.087397

22. *Wang W. [et al]* Malware traffic classification using convolutional neural network for representation learning. International Conference on Information Networking (ICOIN). 2017. P. 712–717. DOI: 10.1109/ICOIN.2017.7899588

23. *Wang W. [et al]* End-to-end encrypted traffic classification with one-dimensional convolution neural networks. IEEE International Conference on Intelligence and Security Informatics (ISI). 2017. P. 43–48. DOI: 10.1109/ISI.2017.8004872

24. *Lotfollahi M. [et al]* Deep packet: A novel approach for encrypted traffic classification using deep learning. Soft Computing. – 2017. – P. 1–14.

25. *Zhang J. [et al]* Robust network traffic classification. IEEE/ACM Transactions on Networking. 2015. V. 23, No 4. P. 1257–1270. DOI: 10.1109/TNET.2014.2320577

26. *Pacheco F. [et al]* Towards the deployment of machine learning solutions in network traffic classification: a systematic survey. IEEE Communications Surveys & Tutorials. 2018. V. 21. No 2. P. 1988–2014. DOI: 10.1109/COMST.2018.2883147

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

28. *Konyshev M. U. [et al]* Formation of probability distributions of binary vectors of the error source of a Markov discrete memory link using the method of "grouping probabilities" of error vectors. Industrial ACS and controllers. 2018. No 3. P. 42.

29. *Konyshev M. U. [et al]* Algorithm for compression of a distribution series of binary multidimensional random variables. Industrial ACS and controllers. 2016. No 8. P. 47–50.

30. Toolkit for the transport layer security and secure sockets layer protocols. Available at: http://openssl.org. (accessed: 14.01.2020).

31. Archive manager WinRAR. Available at: http://rarlab.com (accessed: 14.01.2020).

32. Linux programmer's manual. Available at: http://man7.org/linux/man-pages/man4/random.4.html (accessed: 14.01.2020).

33. Programm environment Anaconda. Available at: https://www.anaconda.com/distribution/ (accessed: 14.01.2020).

34. Breiman, L. Classification and regression trees. Routledge, 2017. 358 p.

**Kozachok Alexander V.** — DSc in Technical Sciences, Russian Federation Security Guard Service Federal Academy.
E-mail: a.kozachok@academ.msk.rsnet.ru
https://orcid.org/0000-0002-6501-2008

**Spirin Andrey A.** — Russian Federation Security Guard Service Federal Academy.
E-mail: spirin_aa@bk.ru
https://orcid.org/0000-0002-7231-5728