ВЕСТНИК
НОВОСИБИРСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

Научный журнал
Основан в ноябре 1999 года

Серия: Информационные технологии 2023. Том 21, № 1

СОДЕРЖАНИЕ

ISSN 1818-7900 (Print). ISSN 2410-0420 (Online)
Вестник НГУ. Серия: Информационные технологии. 2023. Том 21, № 1
Vestnik NSU. Series: Information Technologies, 2023, vol. 21, no. 1

V E S T N I K
NOVOSIBIRSK STATE UNIVERSITY

Scientific Journal
Since 1999, November
In Russan

Series: Information Technologies 2023. Volume 21, № 1

CONTENTS

ISSN 1818-7900 (Print). ISSN 2410-0420 (Online)
Вестник НГУ. Серия: Информационные технологии. 2023. Том 21, № 1
Vestnik NSU. Series: Information Technologies, 2023, vol. 21, no. 1

The journal is published quarterly in Russian since 1999
by Novosibirsk State University Press

The address for correspondence
Faculty of Information Technologies, Novosibirsk State University
1 Pirogov Street, Novosibirsk, 630090, Russia
Tel. +7 (383) 363 42 46

E-mail address: inftech@vestnik.nsu.ru
On-line version: http://elibrary.ru

ISSN 1818-7900 (Print). ISSN 2410-0420 (Online)
Вестник НГУ. Серия: Информационные технологии. 2023. Том 21, № 1
Vestnik NSU. Series: Information Technologies, 2023, vol. 21, no. 1

Научная статья

УДК 519.6
DOI 10.25205/1818-7900-2023-21-1-5-18

Моделирование и количественно-качественный анализ двумерных генераторов хаотических сигналов на основе модулярной арифметики

Алексей Андреевич Гавришев

Национальный исследовательский ядерный университет «МИФИ»

Москва, Россия

alexxx.2008@inbox.ru, https://orcid.org/0000-0002-4242-6152

Аннотация

В данной статье путем совокупного применения программ E&F Chaos, Past, Fractan, Eviews Student Version Lite проведено математическое, численное и компьютерное моделирование некоторых из известных двумерных генераторов хаотических сигналов на основе модулярной арифметики, представленных в работе [4], и осуществлена оценка свойств полученных хаотических сигналов с помощью методов нелинейной динамики (временные и спектральные диаграммы, BDS-статистика, показатель Хёрста). В результате проведенных исследований установлено, что полученные для исследуемых двумерных генераторов хаотических сигналов на основе модулярной арифметики временные и спектральные диаграммы имеют сложный шумоподобный вид, схожий с белым шумом. Полученный диапазон значений BDS-статистики на определенном интервале соответствует белому шуму, а на определенном интервале – персистентным процессам (черный шум). Полученный диапазон значений показателя Хёрста так же находится близко к белому шуму. Полученные результаты показывают, что двумерные генераторы хаотических сигналов на основе модулярной арифметики могут относиться к белому шуму и обладать более выраженными свойствами хаотичности, чем классические генераторы хаотических сигналов, на основе которых они получены. Полученные результаты дополняют и расширяют знания о двумерных генераторах хаотических сигналов на основе модулярной арифметики и открывают широкие перспективы по их использованию в различных практических приложениях.

Ключевые слова

моделирование, методы нелинейной динамики, комплекс программ, хаотические сигналы, модулярная арифметика, количественно-качественная оценка

Для цитирования

Гавришев А. А. Моделирование и количественно-качественный анализ двумерных генераторов хаотических сигналов на основе модулярной арифметики // Вестник НГУ. Серия: Информационные технологии. 2023. Т. 21, № 1. С. 5–18. DOI 10.25205/1818-7900-2023-21-1-5-18

Modeling and Quantitative-Qualitative Analysis of Two-Dimensional Generators of Chaotic Signals Based on Modular Arithmetic

Alexey A. Gavrishev

NRNU MEPhI, Moscow, Russia
alexxx.2008@inbox.ru, https://orcid.org/0000-0002-4242-6152

Annotation

In this article, by the combined application of the programs E&F Chaos, Past, Fractan, Eviews Student Version Lite, mathematical, numerical and computer modeling of some of the well-known two-dimensional generators of chaotic signals based on modular arithmetic presented in [4] was carried out, and the properties of the obtained chaotic signals were evaluated using nonlinear dynamics methods (time and spectral diagrams, BDS-statistics, Hurst exponent). As a result of the conducted research, it was found that the time and spectral diagrams obtained for the studied two-dimensional generators of chaotic signal based on modular arithmetic have a complex noise-like appearance similar to white noise. The resulting range of BDS-statistics values corresponds to white noise at a certain interval, and persistent processes (black noise) at a certain interval. The resulting range of values of the Hurst exponent is also close to white noise. The results obtained show that two-dimensional generators of chaotic signals based on modular arithmetic can relate to white noise and have more pronounced chaotic properties than classical generators of chaotic signals, on the basis of which they are created. The results obtained complement and expand the knowledge about two-dimensional generators of chaotic signals based on modular arithmetic and open up broad prospects for their use in various practical applications.

Keywords

modeling, methods of nonlinear dynamics, software package, chaotic signals, modular arithmetic, quantitative-qualitative assessment

For citation

Gavrishev A. A. Modeling and Quantitative-Qualitative Analysis of Two-Dimensional Generators of Chaotic Signals Based on Modular Arithmetic. *Vestnik NSU. Series: Information Technologies*, 2023, vol. 21, no. 1, pp. 5–18. (in Russ.) DOI 10.25205/1818-7900-2023-21-1-5-18

1. Введение

Открытие нерегулярных хаотических колебаний в детерминированных нелинейных динамических системах различной природы стало одной из крупнейших научных сенсаций конца XX века. Это явление стали называть детерминированным [1]. В последние годы, благодаря исследованиям теории динамического хаоса с помощью быстродействующих компьютеров, стало ясно, что одно из самых важных свойств динамического хаоса – высокая чувствительность к начальным условиям, приводящая к хаотическому поведению во времени, это типичное свойство многих систем. Такое поведение, например, обнаружено в периодических стимулируемых клетках сердца, в электронных цепях, при возникновении турбулентности в жидкостях и газах и др. [2]. В настоящее время концепция динамического хаоса вышла за рамки породившей ее теории нелинейных колебаний и стала новой общенаучной парадигмой. Она легла в основу нового научного направления, называемого синергией. Более того, явление динамического хаоса дало новые важные инженерные идеи, привело к созданию на их основе устройств и теорий, уже активно используемых на практике [1, 2]. Исходя из этого, исследование особенностей динамического хаоса является актуальной научной и практической задачей.

Одним из активно развивающихся направлений теории динамического хаоса является разработка новых способов и методов генерирования хаотических сигналов и исследование их свойств [1–3]. В настоящее время известно достаточно много генераторов хаотических сигналов (ГХС), как построенных на основе классических, так и совершенно новых [1–3]. Вместе с тем имеются направления исследований, которые не проработаны в полней мере. Например,

к таким направлениям относится генерирование хаотических сигналов с помощью двумерных ГХС на основе модулярной арифметики, описанное в работах [4–6].

Приведем пример описания двумерного ГХС на основе модулярной арифметики. Согласно работе [4], двумерные ГХС на основе модулярной арифметики в общем виде возможно описать с помощью выражения (1):

$$M(x, y) = F(x, y) \bmod N, \tag{1}$$

где x и y – это некоторые переменные, N – целое положительное число, являющееся модулем и $F(x, y)$ – это двумерный ГХС, описываемый выражением (2) [4]:

$$F(x,y):\begin{cases} x_{n+1} = G(x_n, y_n) \bmod N, \\ y_{n+1} = G(x_n, y_n) \bmod N. \end{cases} \tag{2}$$

Подставляя выражение (2) в выражение (1), получим окончательное выражение (3) [4]:

$$M(x,y):\begin{cases} x_{n+1} = G(x_n, y_n) \bmod N, \\ y_{n+1} = G(x_n, y_n) \bmod N. \end{cases} \tag{3}$$

С помощью выражений (1) – (3) авторами работы [4], на основе классических двумерных ГХС (Henon Map, Zeraoulia-Sprott Map, Duffing Map), предложены усовершенствованные ГХС (Improved Henon Map, Improved Zeraoulia-Sprott Map, Improved Duffing Map), описываемые выражениями (4) – (6):

$$\begin{cases} x_{n+1} = (1 - ax_n^2 + y_n) \bmod N, \\ y_{n+1} = bx_n \bmod N. \end{cases} \tag{4}$$

$$\begin{cases} x_{n+1} = (1 - ax_n^2 + y_n) \bmod N, \\ y_{n+1} = bx_n \bmod N. \end{cases} \tag{5}$$

$$\begin{cases} x_{n+1} = (1 - ax_n^2 + y_n) \bmod N, \\ y_{n+1} = bx_n \bmod N. \end{cases} \tag{6}$$

Также авторами указанной работы [4] проведены исследования свойств сигналов, получаемых с помощью генераторов, описываемых выражениями (4) – (6). В результате проведенных исследований было установлено, что двумерные ГХС на основе модулярной арифметики обладают следующими свойствами, отличающими их от исходных ГХС [4]:

– значительно большими значениями максимального положительного показателя Ляпунова, энтропии и корреляционной размерности;

– более сложными фазовыми портретами;

– значительно большим диапазоном значений, в которых генерируются хаотические сигналы.

Таким образом, двумерные ГХС на основе модулярной арифметики потенциально обладают более выраженными свойствами хаотичности, чем исходные ГХС. Указанные обстоятельства открывают широкие перспективы по их использованию в различных практических приложения, в частности, в образовательной деятельности, в системах имитационного моделирования, в системах защиты от несанкционированного доступа информации и др. Исходя из этого, требуются дополнительные исследования ГХС указанного класса.

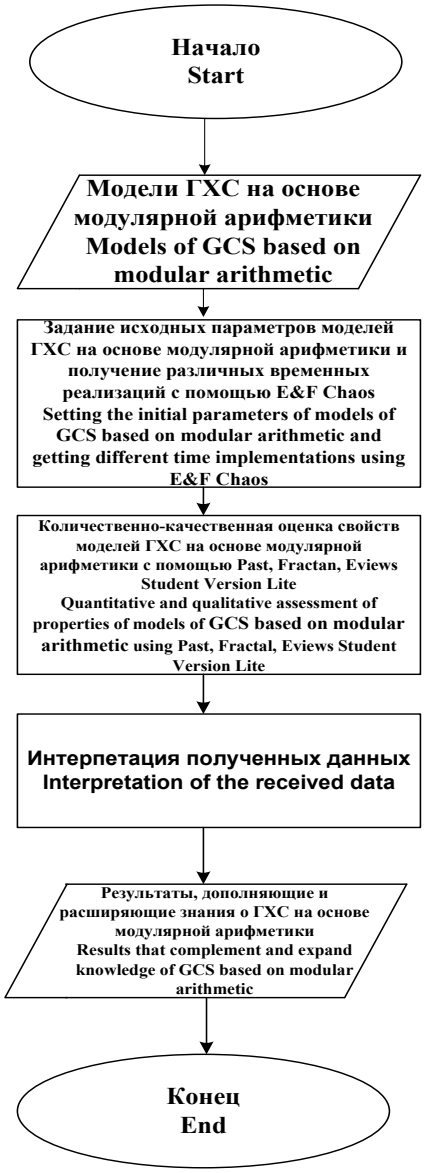
Целью данной статьи является дополнение и расширение знаний о двумерных генераторах хаотических сигналов на основе модулярной арифметики.

Задачей данной статьи является моделирование двумерных генераторов хаотических сигналов на основе модулярной арифметики и оценка свойств получаемых хаотических сигналов с помощью методов нелинейной динамики.

2. Основная часть

2.1. Общий алгоритм осуществляемого в работе математического, численного и компьютерного моделирование, проведенного путем совокупного применения программ

Из литературы известно [7, 8], что в настоящее время перспективным подходом к всестороннему исследованию различных процессов, объектов и явлений выступает использование методов и алгоритмов математического, численного и компьютерного моделирования, которые могут быть успешно реализованы путем совокупного применения различных пакетов программ. В настоящей работе для моделирования двумерных ГХС на основе модулярной арифметики и оценки свойств получаемых хаотических сигналов с помощью методов нелинейной динамики используются следующие известные программы: E&F Chaos, Past, Fractan, Eviews Student Version Lite.



Для удобства процесса моделирования двумерных ГХС на основе модулярной арифметики и оценки свойств получаемых хаотических сигналов с помощью методов нелинейной динамики изобразим используемый в работе комплекс программ в виде блок-схемы. Известно [7, 8], что блок-схемы являются удобным средством изображения различных алгоритмов и получили широкое распространение в научной литературе.

На рис. 1, по аналогии с работами [7, 8], изображена упрощенная блок-схема, описывающая алгоритм осуществляемого в работе математического, численного и компьютерного моделирования, проведенного путем совокупного применения программ E&F Chaos, Past, Fractan, Eviews Student Version Lite. Как видно из упрощенной блок-схемы, представленной на рис. 1, в качестве входных параметров выступают модели двумерных ГХС на основе модулярной арифметики, например, описываемые выражениями (4) – (6). Далее с помощью программы E&F Chaos осуществляется моделирование и выработка различных временных реализаций моделей двумерных ГХС на основе модулярной арифметики. После этого с помощью программ Past, Fractan, Eviews Student Version Lite производится количественно-качественная оценка свойств полученных временных реализаций моделей двумерных ГХС на основе модулярной арифметики. Затем

Рис. 1. Блок-схема алгоритма осуществляемого в работе математического, численного и компьютерного моделирования, проведенного путем совокупного применения программ E&F Chaos, Past, Fractan, Eviews Student Version Lite

Fig. 1. Flowchart of the algorithm of mathematical, numerical and computer modeling carried out in the work, carried out by the combined application of the programs E&F Chaos, Past, Fractan, Eviews Student Version Lite

осуществляется интерпретация полученных данных. Выходными параметрами являются полученные результаты, дополняющие и расширяющие знания о двумерных ГХС на основе модулярной арифметики.

2.2. Обзор некоторых из известных методов нелинейной динамики

Для количественно-качественной оценки свойств временных реализаций, полученных с помощью двумерных ГХС на основе модулярной арифметики, согласно рис. 1, рассмотрим широко распространенные методы нелинейной динамики.

Кратко опишем следующие качественные показатели на основе нелинейной динамики: временные и спектральные диаграммы. Известно [1–3, 9, 10], что произвольный сигнал естественно рассматривать как результат некоторых измерений. Поэтому сигналами чаще всего являются величины, изменяющиеся во времени. Временное представление сигнала в виде функции $s(t)$ или временной диаграммы – это один из самых распространенных способов его описания и исследования. Спектральные диаграммы также являются одним из самых распространенных способов описания и исследования произвольных сигналов. Так, энергетический спектр показывает распределение энергии сигнала по частотам и может быть вычислен, в простейшем случае, с помощью преобразования Фурье [1–3, 10]:

$$S(j\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt. \tag{7}$$

Далее кратко опишем следующие количественные показатели на основе нелинейной динамики, которые не использовались в работе [4] для анализа свойств сигналов, сгенерированных с помощью двумерных ГХС на основе модулярной арифметики: BDS-статистика и показатель Хёрста.

Так, известно [11–13], что BDS-статистика $\overline{w}(\varepsilon)$ базируется на статистических свойствах корреляционной размерности исследуемого процесса в фазовом пространстве, которая в свою очередь определяется корреляционным интегралом. Эти данные дают больше информации о классе процесса (случайные, хаотические, регулярные), чем энергетические показатели. BDS-статистика основана на статистической величине $\overline{w}(\varepsilon)$ [11–13]:

$$w_{m,N}(\varepsilon) = \sqrt{N-m+1}\,\frac{C_{m,N}(\varepsilon) - C_{1,N-m}(\varepsilon)^m}{\sigma_{m,N}(\varepsilon)}, \tag{8}$$

где $C_{m,N}(\varepsilon)$ и $C_{1,N-m}(\varepsilon)$ – корреляционные интегралы, а $\sigma_{m,N}(\varepsilon)$ – среднеквадратическое отклонение.

Задача анализа передаваемого сигнала рассматривается как непараметрическая проверка одной из гипотез: H_0 – наблюдаемые данные $\overline{x} = (x_1,...,x_n)$ независимы и одинаково распределены (белый шум) и H_1 – данные не относятся к белому шуму, что возможно в случае, когда они являются смесью шума и сигнала. В качестве теста на достоверность гипотезы H_0 об отсутствии в наблюдении хаотического процесса принимается выполнение неравенства $w_{m,N}(\varepsilon) \le |1{,}96|$, для значения статистики $w_{m,N}(\varepsilon)$, что соответствует уровню значимости $\alpha = 0{,}05$, тогда с 95 % вероятностью можно принять гипотезу H_0 (белый шум). Критическая область уровня $\alpha = 0{,}05$ состоит из двух бесконечных полуинтервалов $(-\infty, -1{,}96]$ и $[1{,}96, \infty)$. В отсутствие шумов наблюдения применение критерия значимости к статистике $w_{m,N}(\varepsilon)$позволяет эффективно решать задачу по классификации наблюдения ($w_{m,N}(\varepsilon) \le |1{,}96|$) [11–13].

Как известно [9, 13–15], показатель Хёрста H позволяет разделить между собой периодические и случайные процессы. Показатель Хёрста H описывается выражением, представленным формулой (1) [9, 13–15]:

$$R/S = (\tau/2)^H, \quad (9)$$

где R – нормированный размах вариации (разность максимального и минимального значений измеряемого параметра), S – стандартное отклонение (корень квадратный от дисперсии), τ – период (длина ряда) наблюдений.

В соответствии с [9, 13–15], значения показателя Хёрста $0 < H < 0{,}50$ типичны для так называемых «антиперсистентных процессов» (эргодические ряды), значения $0{,}50 < H < 1$ характерны для систем, в которых имеется та или иная форма упорядоченности, а значение $H \approx 0{,}50$ соответствует понятию белого шума.

2.3. Применение комплекса программ для моделирования и анализа модели двумерного ГХС на основе модулярной арифметики, описываемого выражением (4)

Основываясь на алгоритме, приведенном на рис. 1, проведем моделирование двумерного ГХС на основе модулярной арифметики и оценку полученных хаотических сигналов. В качестве двумерного ГХС на основе модулярной арифметики выберем Improved Henon Map, описываемый выражением (4). Моделирование проведем с помощью программы E&F Chaos [16]. Для двумерного ГХС на основе модулярной арифметики, описываемого выражением (4), в соответствии с работой [4], рассмотрим следующие случаи:

1) $a = 6$ и $b \in [6 \div 105]$;
2) $b = 6$ и $a \in [6 \div 105]$;
3) $a \in [6 \div 105]$ и $b \in [6 \div 105]$.

Рассмотрим первый случай, когда $a = 6$ и $b \in [6 \div 105]$. В соответствии с указанными значениями параметров, при изменении значения переменной b в диапазоне примерно от 6 до 105 с шагом 1 были получены различные временные реализации. Длина каждой из реализаций – 2000. В частности, на рис. 2 представлен пример временной диаграммы, полученной с помощью программы PAST [9, 17]. Другие полученные временные диаграммы имеют схожий вид.

Соответствующая рис. 2 спектральная диаграмма представлена на рис. 3. Спектральные диаграммы получены с помощью программы PAST [9, 17]. Другие полученные спектральные диаграммы имеют схожий вид.

Анализ полученных качественных показателей, представленных на рис. 2, 3, проведен в разделе 2.4.

Далее проведем расчеты BDS-статистики, согласно формуле (8), и показателя Хёрста, согласно формуле (9), для полученных данных (табл. 1). Расчеты BDS-статистики авторами проведены в программе Eviews Student Version Lite [18]. Расчеты показателя Хёрста авторами проведены в программе Fractan [9, 19].

Таблица 1

Расчеты для модели, описываемой выражением (4),
для первого рассматриваемого случая

Table 1

Calculations for a Model Described by the Expression (4)
for the First Case Under Consideration

Название показателя Indicator name	Значение показателя The value of the indicator
BDS-статистика BDS-statistic	0,70÷3,80
Показатель Хёрста Hurst exponent	0,40÷0,55

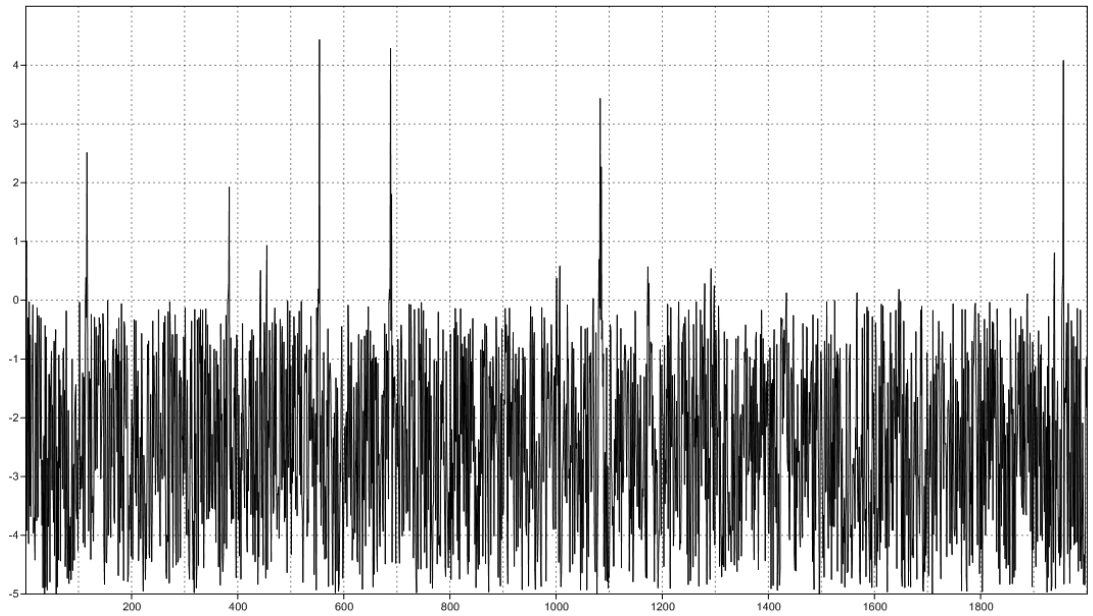


Рис. 2. Пример временной диаграммы для модели, представленной выражением (4), для первого рассматриваемого случая
Fig. 2. Example of a time diagram for a model represented by the expression (4) for the first case under consideration

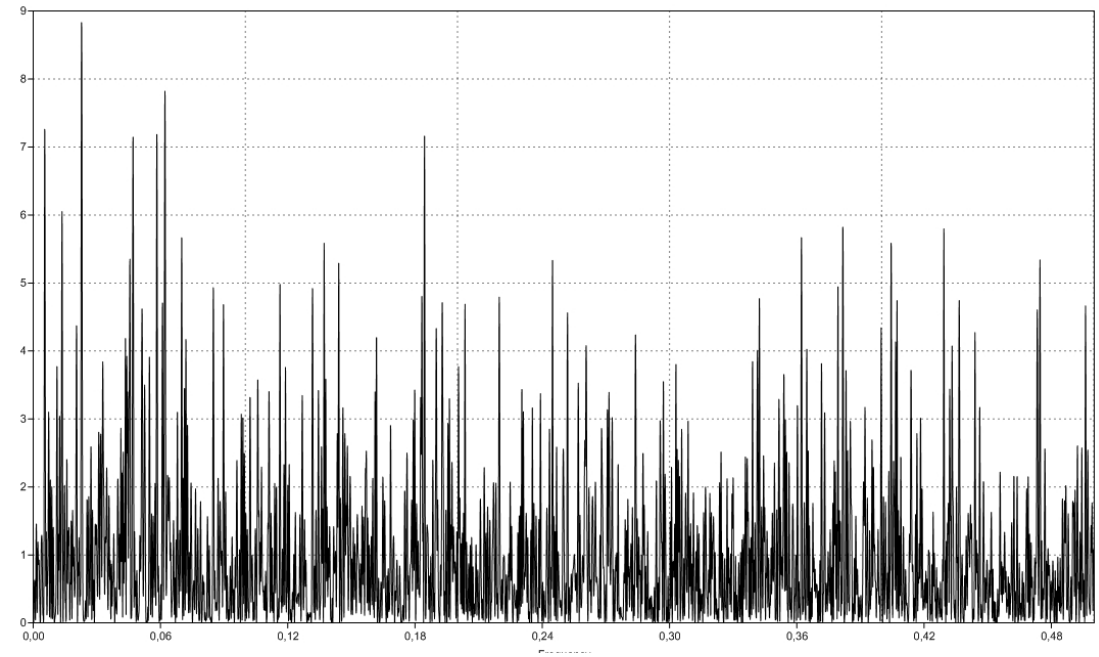


Рис. 3. Пример спектральной диаграммы для модели, представленной выражением (4), для первого рассматриваемого случая
Fig. 3. Example of a spectral diagram for a model represented by the expression (4) for the first case under consideration

Анализ полученных количественных показателей, представленных в табл. 3, проведен в разделе 2.4.

Рассмотрим второй случай, когда $b = 6$ и $a \in [6 \div 105]$. В соответствии с указанными значениями параметров при изменении значения переменной a в диапазоне примерно от 6 до 105 с шагом 1 были получены различные временные реализации. Длина каждой из реализаций – 2000. В частности, на рис. 4 представлен пример временной диаграммы, полученной с помощью программы PAST [9, 17]. Другие полученные временные диаграммы имеют схожий вид.

Соответствующая рис. 4 спектральная диаграмма представлена на рис. 5. Спектральные диаграммы получены с помощью программы PAST [9, 17]. Другие полученные спектральные диаграммы имеют схожий вид.

Анализ полученных качественных показателей, представленных на рис. 4, 5, проведен в разделе 2.4.

Далее проведем расчеты BDS-статистики, согласно формуле (8), и показателя Хёрста, согласно формуле (9), для полученных данных (таб. 2). Расчеты BDS-статистики авторами проведены в программе Eviews Student Version Lite [18]. Расчеты показателя Хёрста авторами проведены в программе Fractan [9, 19].

Таблица 2

Расчеты для модели, описываемой выражением (4),
для второго рассматриваемого случая

Table 2

Calculations for a Model Described by the Expression (4)
for the Second Case Under Consideration

Название показателя Indicator name	Значение показателя The value of the indicator
BDS-статистика BDS-statistic	0,30÷4,60
Показатель Хёрста Hurst exponent	0,45÷0,65

Анализ полученных количественных показателей, представленных в таб. 2, проведен в разделе 2.4.

Рассмотрим третий случай, когда $a \in [6 \div 105]$ и $b \in [6 \div 105]$. В соответствии с указанными значениями параметров, при изменении значений переменных a и b в диапазоне примерно от 6 до 105 с шагом 1 были получены различные временные реализации. В частности, на рис. 6 представлен пример временной диаграммы, полученной с помощью программы PAST [9, 17]. Другие полученные временные диаграммы имеют схожий вид.

Соответствующая рис. 6 спектральная диаграмма представлена на рис. 7. Спектральные диаграммы получены с помощью программы PAST [9, 17]. Другие полученные спектральные диаграммы имеют схожий вид.

Анализ полученных качественных показателей, представленных на рис. 6, 7, проведен в разделе 2.4.

Далее проведем расчеты BDS-статистики, согласно формуле (8), и показателя Хёрста, согласно формуле (9), для полученных данных (табл. 3). Расчеты BDS-статистики авторами проведены в программе Eviews Student Version Lite [18]. Расчеты показателя Хёрста авторами проведены в программе Fractan [9, 19].

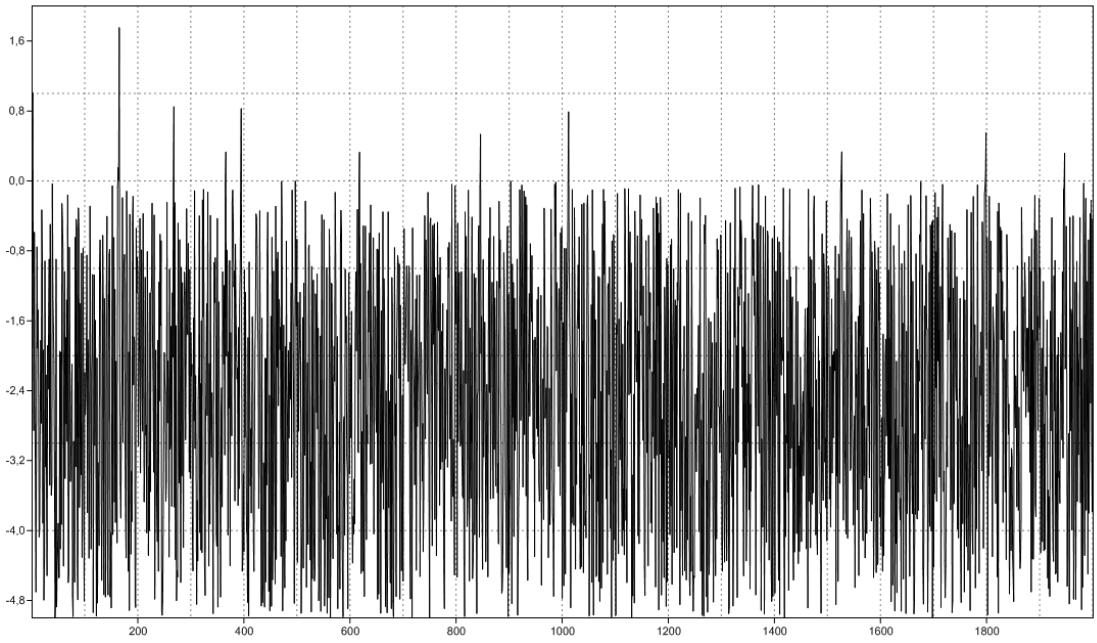


Рис. 4. Пример временной диаграммы для модели, представленной выражением (4), для второго рассматриваемого случая
Fig. 4. Example of a time diagram for a model represented by the expression (4) for the second case under consideration

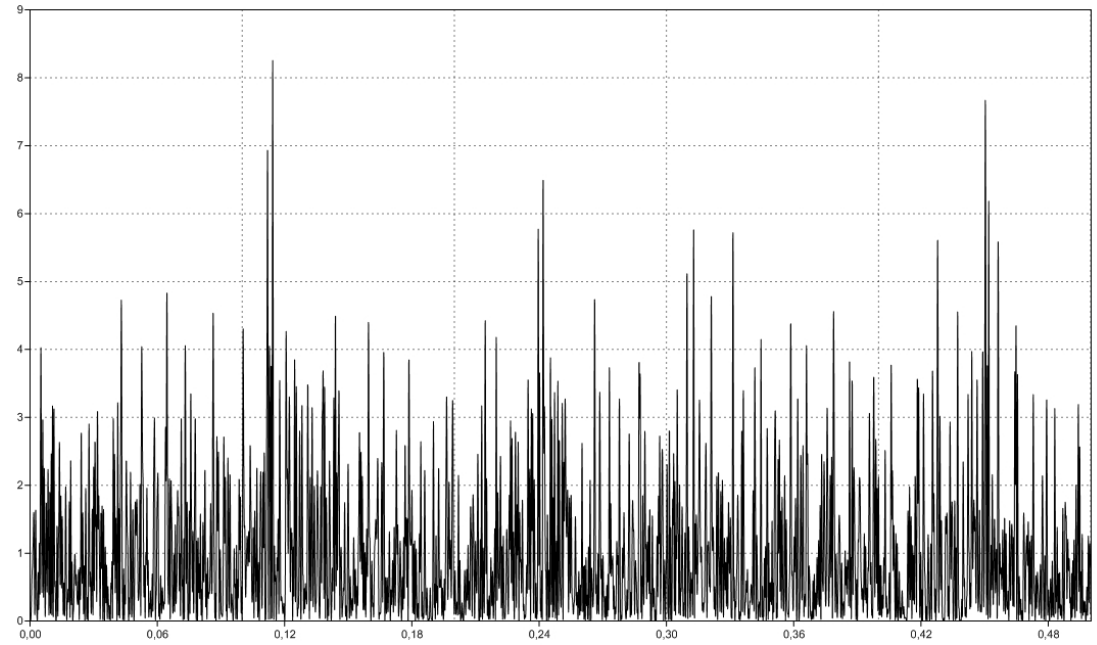


Рис. 5. Пример спектральной диаграммы для модели, представленной выражением (4), для второго рассматриваемого случая
Fig. 5. Example of a spectral diagram for a model represented by the expression (4) for the second case under consideration

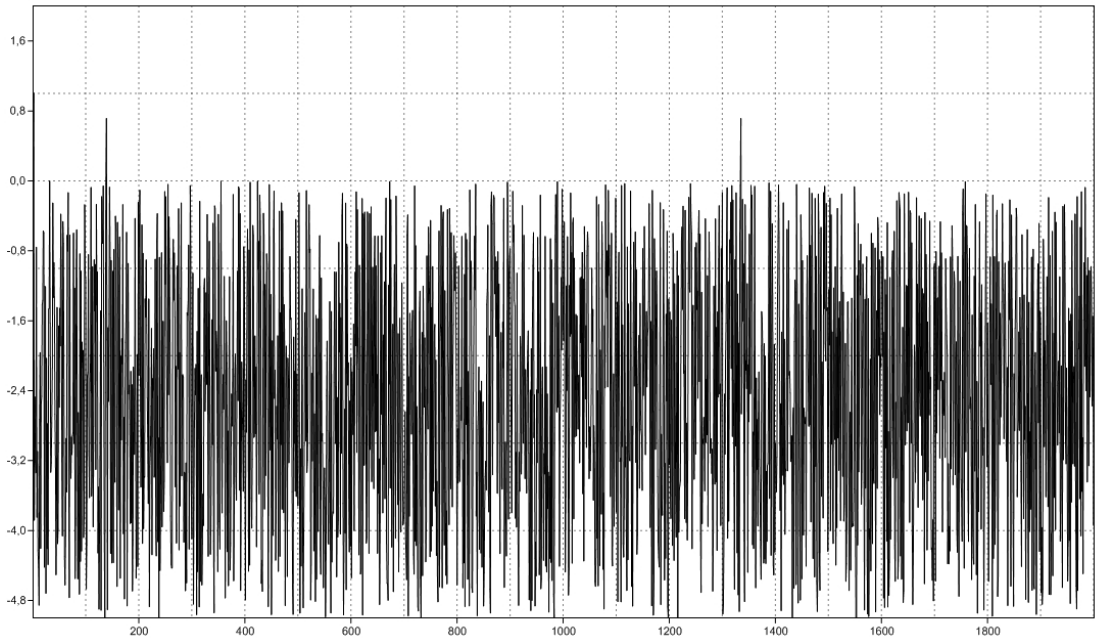


Рис. 6. Пример временной диаграммы для модели, представленной выражением (4), для третьего рассматриваемого случая
Fig. 6. Example of a time diagram for a model represented by the expression (4) for the third case under consideration

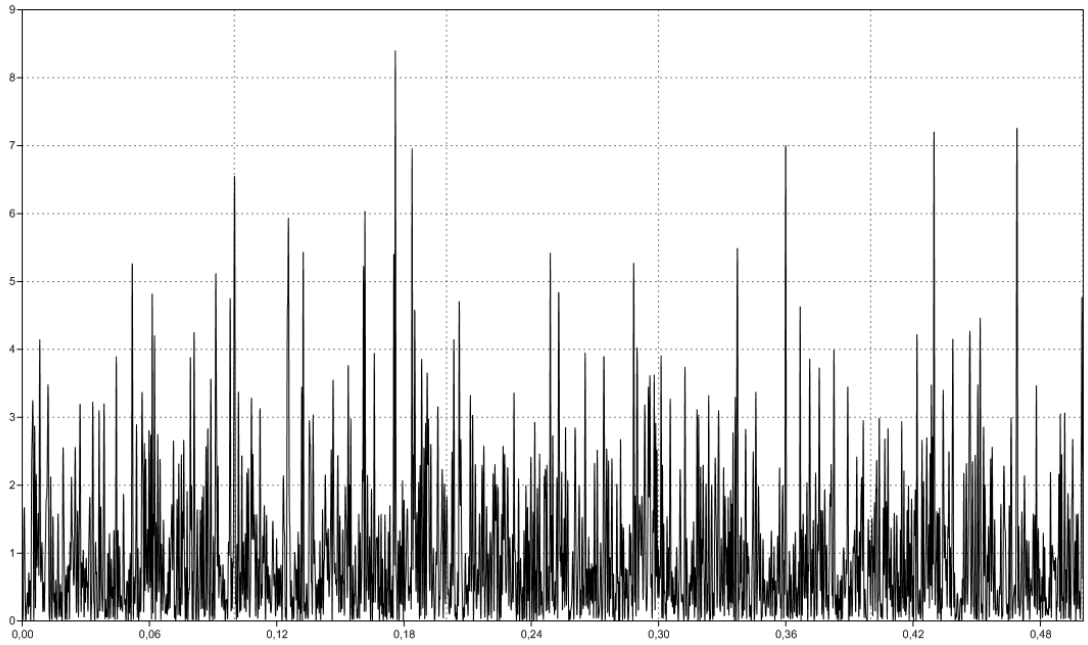


Рис. 7. Пример спектральной диаграммы для модели, представленной выражением (4), для третьего рассматриваемого случая
Fig. 7. Example of a spectral diagram for a model represented by the expression (4) for the third case under consideration

Таблица 3

Расчеты для модели, описываемой выражением (4),
для третьего рассматриваемого случая

Table 3

Calculations for a Model Described by the Expression (4)
for the Third Case Under Consideration

Название показателя Indicator name	Значение показателя The value of the indicator
BDS-статистика BDS-statistic	0,50÷3,00
Показатель Хёрста Hurst exponent	0,42÷0,62

Анализ полученных количественных показателей, представленных в табл. 3, проведен в разделе 2.4.

2.4. Интерпретация полученных данных для исследуемых двумерных ГХС на основе модулярной арифметики

Далее, основываясь на алгоритме, приведенном на рис. 1, осуществим интерпретацию полученных данных для двумерного ГХС на основе модулярной арифметики, описываемого выражением (4).

На основе работ [1–3] проведем интерпретацию полученных качественных показателей. Полученные временные диаграммы, представленные на рис. 2, 4, 6, имеют сложный шумоподобный вид и являются непрерывными во временной области. Это указывает на то, что исследуемые сигналы могут обладать свойствами хаотичности. Полученные спектральные диаграммы, представленные на рис. 3, 5, 7, имеют сложный шумоподобный вид, схожий с белым шумом. Это указывает на то, что исследуемые сигналы могут обладать свойствами хаотичности.

На основе работ [9, 13–15, 19] проведем интерпретацию полученных количественных показателей, представленных в табл. 1–3. Согласно работам [13–15], полученный для двумерного ГХС на основе модулярной арифметики, описываемого выражением (4), диапазон значений BDS-статистики $\overline{w}(\varepsilon) \in [0{,}30 \div 4{,}60]$, на определенном интервале соответствует белому шуму ($w_{m,N}(\varepsilon) \le |1{,}96|$), а на определенном интервале – персистентным процессам (черный шум) ($\overline{w}(\varepsilon) \in [1{,}70 \div 5{,}00]$). Таким образом, полученные временные реализации двумерного ГХС на основе модулярной арифметики находятся достаточно близко к границе диапазона, определяющего белый шум, по сравнению с исходным ГХС Henon Map [20], диапазон значений BDS-статистики которого равен $\overline{w}(\varepsilon) \in [21{,}90 \div 45{,}00]$.

Согласно работам [9, 15, 19], полученный для двумерного ГХС на основе модулярной арифметики, описываемого выражением (4), диапазон значений показателя Хёрста $H \in [0{,}40 \div 0{,}65]$ также находится близко к белому шуму ($H \approx 0{,}50$) и совпадает с исходным ГХС Henon Map [20], диапазон значений показателя Хёрста которого равен $H \in [0{,}46 \div 0{,}52]$.

Таким образом, временные и спектральные диаграммы, BDS-статистика и показатель Хёрста показывают, что двумерный ГХС на основе модулярной арифметики, представленный выражением (4), может относиться к белому шуму и обладает более выраженными свойствами хаотичности, чем исходный ГХС, на основе которого он получен. Полученный результат в целом совпадает с результатами из работы [4].

Заключение

Таким образом, в данной статье на основе осуществленного математического, численного и компьютерного моделирования, проведенного путем совокупного применения программ E&F Chaos, Past, Fractan, Eviews Student Version Lite, установлено, что:

1) полученные временные и спектральные диаграммы двумерного ГХС на основе модулярной арифметики, описываемого выражением (4), имеют сложный шумоподобный вид, схожий с белым шумом;

2) полученный для двумерного ГХС на основе модулярной арифметики, описываемого выражением (4), диапазон значений BDS-статистики $\overline{w}(\varepsilon) \in [0{,}30 \div 4{,}60]$ на определенном интервале соответствует белому шуму ($w_{m,N}(\varepsilon) \le |1{,}96|$), а на определенном интервале – персистентным процессам (черный шум) ($\overline{w}(\varepsilon) \in [1{,}70 \div 5{,}00]$);

3) полученный для двумерного ГХС на основе модулярной арифметики, описываемого выражением (4), диапазон значений показателя Хёрста $H \in [0{,}40 \div 0{,}65]$ также находится близко к белому шуму ($H \approx 0{,}50$);

4) полученные результаты показывают, что двумерные ГХС на основе модулярной арифметики могут относиться к белому шуму и обладать более выраженными свойствами хаотичности, чем классические ГХС, на основе которых они созданы. Полученный результат в целом совпадает с результатами из известных работ;

5) полученные результаты дополняют и расширяют знания о двумерных ГХС на основе модулярной арифметики и открывают широкие перспективы по их использованию в различных практических приложениях.

Список литературы

1. **Шахтарин Б. И. и др.** Генераторы хаотических колебаний. М.: Горячая линия – Телеком, 2014. 248 с.
2. **Шустер Г.** Детерминированный хаос: Введение. М.: Мир. 1988. 240 с.
3. **Kehui Su**n Chaotic Secure Communication: Principles and Technologies. Tsinghua University Press and Walter de Gruyter GmbH. 2016. 333 p.
4. **Zhongyun Hua, Yinxing Zhang, Yicong** Zhou Two-Dimensional Modular Chaotification System for Improving Chaos Complexity // IEEE Transactions on signal processing. 2020. V. 68. Pp. 1937-1948. DOI: 10.1109/TSP.2020.2979596.
5. **Moysis L., Kafetzis I., Baptista M.S., Volos C.** Chaotification of One-Dimensional Maps Based on Remainder Operator Addition // Mathematics. 2022. No. 10. Pp. 2801. DOI: 10.3390/math10152801.
6. **Hua Zhong Yun, Zhou Bing Hang, Zhang Yin Xing, Zhou Yi Cong** Modular chaotification model with FPGA implementation // Science China. 2021 V. 64 No. 7. Pp. 1472–1484. DOI: 10.1007/s11431-020-1717-1.
7. **Нарожнов В. В.** Моделирование нелинейного осциллятора при наличии упругих соударений: дис. ... канд. техн. наук: 05.13.18. Нальчик, 2018. 134 с.
8. **Гавришев А. А.** Применение методов нелинейной динамики для количественно-качественной оценки свойств 2D-моделей S-хаоса // Прикладная информатика. 2021. Т. 16. № 1 (91). С. 125-143. DOI: 10.37791/2687-0649-2021-16-1-125-143.
9. **Гавришев А. А., Жук А. П.** Применение методов нелинейной динамики для исследования хаотичности сигналов-переносчиков защищенных систем связи на основе динамического хаоса // Вестник НГУ. Серия: Информационные технологии. 2018. Т. 16. № 1. С. 50–60. DOI: 10.25205/1818-7900-2018-16-1-50-60.
10. **Каратаева Н. А.** Радиотехнические цепи и сигналы. Ч. 1. Томск: Томский межвузовский центр дистанционного образования, 2012. 260 с.

11. **Васюта К. С.** Классификация процессов в инфокоммуникационных радиотехнических системах с применением BDS-статистики // Проблемы телекоммуникаций. 2012. № 4. С. 63–71.
12. **Васюта К. С.** Новый подход к оценке параметров хаотических сигналов, наблюдаемых на фоне шума, с использованием «нелинейной динамической статистики» // Проблемы телекоммуникаций. 2010. № 1. С. 109–114.
13. **Гавришев А. А.** Моделирование и количественно-качественный анализ распространенных защищенных систем связи // Прикладная информатика. 2018. Т. 13. № 5 (77). С. 84–122.
14. **Карманов А. П., Кочева Л. С., Щемелинина Т. Н.** Применение методов нелинейной динамики для анализа результатов мониторинга сточных вод // Известия высших учебных заведений. Лесной журнал. 2014. № 6. С. 129–137.
15. **Петрунина Т. С.** Численный анализ структурных свойств хаотических временных рядов // Вестник Нац. техн. ун-та «ХПИ». Темат. вып.: Системный анализ, управление и информационные технологии. 2011. № 32. С. 71–75.
16. **Diks C., Hommes C., Panchenko V. et al.** E&F Chaos: A User Friendly Software Package for Nonlinear Economic Dynamics // Comput Econ. 2008. No. 32. Pp. 221–244 DOI: 10.1007/s10614-008-9130-x.
17. **Hammer O., Harper D. A. T.** Paleontological Data Analysis. Blackwell Publ., 2006. 370 p.
18. **Гавришев А. А., Жук А. П.** Применение программы EViews для анализа защищенных систем связи на основе хаотических сигналов на основе BDS-статистики // T-Comm: Телекоммуникации и транспорт. 2018. Т. 12. № 11. С. 43–50. DOI 10.24411/2072-8735-2018-10175.
19. **Kovalenko A. N.** Fractal characterization of nanostructured materials // Nanosystems: physics, chemistry, mathematics. 2019. No. 10 (1). Pp. 42–49. DOI: 10.17586/2220-8054-2019-10-1-42-49.
20. **Гавришев А. А., Осипов Д. Л.** Развитие использования методов нелинейной динамики для обнаружения радиосигналов с псевдослучайной перестройкой рабочей частоты, используемых в каналах связи беспилотных летательных аппаратов // Сибирский пожарно-спасательный вестник. 2021. № 4(23). С. 92–96. DOI: 10.34987/vestnik.sibpsa.2021.71.49.011.

References

1. **Shahtarin B. I. et al.** Generators of chaotic oscillations: a tutorial. Moscow. Goryachaya liniya-Telekom Publ. 2014. 248 p.
2. **Shuster G.** [Deterministic chaos: Introduction]. Moscow. Mir Publ. 1988. 240 p.
3. **Sun K.** Chaotic Secure Communication: Principles and Technologies. Tsinghua University Press and Walter de Gruyter GmbH, 2016. 333 p.
4. **Hua Z., Zhang Y., Zhou Y.** Two-Dimensional Modular Chaotification System for Improving Chaos Complexity // IEEE Transactions on signal processing. 2020. Vol. 68. Pp. 1937–1948. DOI 10.1109/TSP.2020.2979596
5. **Moysis L., Kafetzis I., Baptista M. S., Volos C.** Chaotification of One-Dimensional Maps Based on Remainder Operator Addition. Mathematics. 2022. No. 10. P. 2801. DOI 10.3390/math10152801
6. **Hua Z. Y., Zhou B. H., Zhang Y. X., Zhou Y.** Cong Modular chaotification model with FPGA implementation. Science China. 2021. Vol. 64, no. 7. Pp. 1472–1484. DOI 10.1007/s11431-020-1717-1
7. **Narozhnov V. V.** Modeling of a nonlinear oscillator in the presence of elastic collisions]: dis. ... cand. tech. sciences. Nal'chik, 2018, 134 p.
8. **Gavrishev A.** Application of nonlinear dynamics methods for quantitative and qualitative evaluation of properties of 2D models of S-chaos // Journal of Applied Informatics. 2021. Vol. 16, no.1. Pp.125–143. (in Russ.) DOI 10.37791/2687-0649-2021-16-1-125-143

9. **Gavrishev A. A., Zhuk A. P.** Application of Methods of Nonlinear Dynamics to Study the Chaotic State of the Carrier Signals of Secure Communication Systems Based on Dynamic Chaos. Vestnik NSU. Series: Information Technologies. 2018. Vol. 16, no. 1. Pp. 50–60. (in Russ.) DOI 10.25205/1818-7900-2018-16-1-50-60
10. **Karataeva N.A.** Radio engineering circuits and signals. P. 1. Tomsk: Tomsk interuniversity center for distance education Publ., 2012. 260 p.
11. **Vasyuta C. S.** Classification of process in infocomunication radiotehnic systems using BDS-statistics. Problemy telekomunikatsiy. 2012. No. 4. Pp. 63–71. (in Russ.)
12. **Vasyuta C. S.** A new approach to estimation of the parameters of chaotic signals observed on the background noise, using the "nonlinear dynamic statistics". Problemy telekomunikatsiy. 2010. No. 1. Pp. 109–114 (in Russ.)
13. **Gavrishev A. A.** Modeling and quantitative and qualitative analysis of common secure communication systems. Journal of Applied Informatics. 2018. Vol. 13, no. 5. Pp. 84–122. (in Russ.)
14. **Karmanov A. P., Kocheva L. S., Shchemelinina T. N.** Application of Non-Linear Dynamics Methods for Analysis of Results of Industrial Wastewater. Forestry journal. 2014. No. 6. Pp. 129–137. (in Russ.)
15. **Petrunina T. S.** Numerical analysis of the structural properties of the chaotic time series. Vestnik Nats. tekhn. unta «KhPI». Temat. vyp.: Sistemnyy analiz, upravlenie i informatsionnye tekhnologii – Bulletin of NTU "KhPI". The themed slots. vol.: System analysis, management and information technology. 2011. No. 32. Pp. 71–75.
16. **Diks C., Hommes C., Panchenko V. et al.** E&F Chaos: A User Friendly Software Package for Nonlinear Economic Dynamics. Comput. Econ. 2008. No. 32. Pp. 221–244. DOI 10.1007/s10614-008-9130-x
17. **Hammer O., Harper D. A. T.** Paleontological Data Analysis. Blackwell Publ., 2006. 370 p.
18. **Gavrishev A. A., Zhuk A. P.** Application of the Eviews program for the analysis of secure communication systems based on chaotic signals based on BDS-statistics. T-Comm. 2018. Vol. 12, no.11. Pp. 43–50. (in Russ.) DOI 10.24411/2072-8735-2018-10175
19. **Kovalenko A. N.** Fractal characterization of nanostructured materials. Nanosystems: physics, chemistry, mathematics. 2019. No. 10(1). Pp. 42–49. DOI 10.17586/2220-8054-2019-10-1-42-49
20. **Gavrishev A. A., Osipov D. L.** Application of nonlinear dynamics methods for detecting radio signals with frequency-hopping spread spectrum used in communication channels of unmanned aerial vehicles. Siberian Fire and Rescue Bulletin. 2021. No. 4(23). Pp. 92–96. (in Russ.) DOI 10.34987/vestnik.sibpsa.2021.71.49.011

Сведения об авторах

Гавришев Алексей Андреевич, магистрант, Институт интеллектуальных и кибернетических систем, НИЯУ «МИФИ», (Москва, Россия)

Information about the Authors

Aleksey A. Gavrishev, Master's Student, Institute of Intelligent and Cybernetic Systems, NRNU MEPhI (Moscow, Russia)

Статья поступила в редакцию 17.01.2023;
одобрена после рецензирования 20.03.2023; принята к публикации 20.03.2023
The article was submitted 17.01.2023;
approved after reviewing 20.03.2023; accepted for publication 20.03.2023

Научная статья

УДК 004.056
DOI 10.25205/1818-7900-2023-21-1-19-31

Анализ рисков нарушения информационной безопасности от компьютерных атак при нарастающей величине потерь от их реализации

Алла Юрьевна Ермакова[1]
Алексей Борисович Лось[2]

[1]МИРЭА – Российский технологический университет
Москва, Россия

[2]Национальный исследовательский университет «Высшая школа экономики»
Москва, Россия

[1]ermakova_a@mirea.ru

[3]alos@hse.ru

Аннотация

В статье рассматриваются вопросы анализа рисков нарушения информационной безопасности в случае необходимости учета суммарного ущерба при возникновении компьютерных инцидентов.

В качестве математической модели возникновения инцидентов рассматривается дискретная временная модель, при которой инциденты в информационной системе возникают в случайные дискретные моменты времени. Под инцидентами при этом понимаются как непреднамеренные события, такие как сбои в работе, нарушения правил эксплуатации, так и преднамеренные – компьютерные атаки, попытки несанкционированного доступа и аналогичные ситуации. В случае применения риск-ориентированного подхода считаем, что наступление каждого инцидента сопровождается ущербом, величина которого фиксирована. Настройка реагирования на инциденты рассмотрена в двух основных сценариях с накоплением потерь и без таковой.

В работе в рамках рассматриваемых сценариев оцениваются риски нарушения информационной безопасности, в частности, найдено вероятностное распределение времени безопасной работы информационной системы. В качестве иллюстрации рассматриваемого подхода построены прогнозные модели количества несанкционированных операций со счетами юридических лиц и количества несанкционированных операций с использованием платежных карт. Рассматриваемые модели строятся по реальным данным об инцидентах, с применением разработанного ранее метода прогнозирования на основе непрерывной аппроксимирующей функции.

Ключевые слова

модель управления рисками, инциденты информационной безопасности, ущерб информационным активам, защищенность информационной системы, прогнозирование инцидентов.

Для цитирования

Ермакова А. Ю., Лось А. Б. Анализ рисков нарушения информационной безопасности от компьютерных атак при нарастающей величине потерь от их реализации// Вестник НГУ. Серия: Информационные технологии. 2023. Т. 21, № 1. С. 19–31. DOI 10.25205/1818-7900-2023-21-1-19-31

Analysis of the Risks of Information Security Violations from Computer Attacks with an Increasing amount of Damage from Their Implementation

Alla Yu. Ermakova[1], Alexey B. Los[2]

[1]Russian Technological University
Moscow, Russian Federation

[2]HSE University
Moscow, Russian Federation

[1]ermakova_a@mirea.ru
[3]alos@hse.ru

Abstract

The article discusses the issues of analyzing the risks of information security violations if it is necessary to take into account the total damage in the event of computer incidents.

A discrete time model is considered as a mathematical model of the occurrence of incidents, in which incidents in an information system occur at random discrete moments of time. Incidents are understood as unintended events, such as malfunctions, violations of operating rules, and intentional – computer attacks, unauthorized access attempts and similar situations. In the case of a risk-based approach, we believe that the occurrence of each incident is accompanied by damage, the magnitude of which is fixed.

The use of protective equipment reduces the likelihood of the risk of incidents. But setting up an incident response can be implemented in one of two main scenarios.

In one variant, when another incident occurs, the corresponding amount of damage is compared with its maximum allowable value. If the amount of damage received, regardless of the damage caused by other incidents, does not exceed the specified threshold, then the information system continues to operate normally. Otherwise, the security policy is reviewed, the necessary additional protective measures are introduced and other measures are taken to improve the security of the information system.

In another variant of the scenario, when incidents occur that lead to a violation of information security, the values of damages from successive incidents are summed up and the value of the sum is compared with the maximum allowable amount of damage. If, during the next incident, the total damage from all previous incidents does not exceed the maximum set value, the information system continues to operate normally. Otherwise, it is concluded that it is necessary to introduce additional protection measures.

In the work, within the framework of the models under consideration, the risks of information security violations are assessed, in particular, the probabilistic distribution of the time of safe operation of the information system is found. As an illustration of the considered approach, predictive models of the number of unauthorized transactions with accounts of legal entities and the number of unauthorized transactions using payment cards are constructed. The models under consideration are based on real data about incidents, using a previously developed forecasting method based on a continuous approximating function.

Keywords

risk management model, information security incidents, property damage to information assets, information system security, predicting incidents.

For citation

Ermakova A.Yu., Los A. B. Analysis of the Risks of Information Security Violations from Computer Attacks with an Increasing amount of Damage from Their Implementation. *Vestnik NSU. Series: Information Technologies*, 2023, vol. 21, no. 1, pp. 19–31. (in Russ.) DOI 10.25205/1818-7900-2023-21-1-19-31

Введение

В данной работе рассматривается актуальная задача разработки теоретико-вероятностных моделей, применяемых при оценке уровня защищенности информационных систем. Процедура защиты информационной системы (далее – ИС) направлена прежде всего на предотвращение возникновения ситуаций, нарушающих штатный режим работы, в частности, компьютерных атак злоумышленников, ошибок в действиях персонала, сбоев в работе оборудования и других аналогичных событий. Компьютерные атаки на информационные системы различных

организаций давно уже стали одной из главных проблем в их работе и причиняют значительный ущерб в различных вариантах, начиная с хищений денежных средств и заканчивая потерей репутации и различной клиентуры. В связи с возрастанием интенсивности компьютерных атак и расширением их спектра возникает вопрос оценки уровня защищенности конкретной информационной системы. Различным подходам к решению данной задачи посвящено много работ в научной литературе [1–3].

В большинстве публикуемых работ аналитическое прогнозирование рисков предлагается осуществлять на основе вероятностного моделирования исследуемых систем [4–6]. Модели процесса проведения компьютерных атак, модели их количественного оценивания и вопросы оценки защищенности информационных систем исследовались в работах [7–9].

В основных нормативных документах по защите информации подход к данной проблеме основан на оценке рисков, возникающих при появлении различных инцидентов в информационной системе, которые приводят к нарушению штатного режима ее работы. При этом функция риска вычисляется с учетом вероятности появления данных инцидентов и ущерба, возникающего от их реализации. Принятие решения о защищенности системы будет в случае, когда значение риска не превышает максимального уровня ущерба.

Однако, как было отмечено ранее в работах [10–15], такой подход позволяет сделать вывод о достаточности мер защиты системы только на момент исследования, и не дает возможности оценить временной промежуток, в течение которого будет обеспечиваться защищенность информационной системы.

Ранее авторами предлагался подход к оценке уровня защищенности информационной системы, основанный на построении временных моделей компьютерных атак и моделей возможных потерь от их реализации.

Смысл такого подхода состоит во введении зависимости от времени t величин, входящих в формулу риска: $p(y_i)$ – вероятности наступления инцидента (реализации угрозы) y_i и u_i – величины ущерба:

$$p(y_i) = p_{y_i}(t),\ u_i = u_i(t).$$

В этом случае функция рисков R также становится функцией от времени t:

$$R = R(t) = \sum_{i=1}^{n} p_{y_i}(t) \cdot u_i(t).$$

Поскольку функции $p_{y_i}(t)$ и $u_i(t)$, как правило, являются неубывающими функциями времени t, нетрудно видеть, что функция рисков $R(t)$ также будет неубывающей функцией времени t.

В этом случае для оценки времени безопасной работы информационной системы можно составить уравнение относительно неизвестного переменного t:

$$R(t) = \sum_{i=1}^{n} p_{y_i}(t) \cdot u_i(t) = R_0.$$

Обозначим через T_0 корень данного уравнения, при котором

$$R(t) - R_0 = 0. \quad (1)$$

Заметим, что в данной модели величина T_0 является временным промежутком, в течение которого ущерб ИС при реализации возникающих угроз достигнет предельно допустимого значения и эксплуатация ИС, согласно введенной модели, перестанет быть безопасной.

В таком случае целесообразно рассматривать величину T_0 как объективную характеристику защищенности ИС. В рамках такого подхода основной задачей становится разработка способов построения непрерывных функций $p_{y_i}(t)$ и $u_i(t)$.

При наличии объективных данных об имевших место инцидентах, которыми обычно располагают службы защиты информации в организациях, одним из возможных подходов к построению непрерывных функций $p_{y_i}(t)$ и $u_i(t)$ может быть разработанный ранее и подробно изложенный в работах [11–13], основанный на построении функций метод прогнозирования состояний динамических систем, наиболее близко расположенных от имеющихся значений параметров состояний указанных систем.

Далее представлены результаты экспериментов по построению функций прогнозирования потерь от хищений со счетов юридических лиц с использованием платежных карт и вычислению на их основе величины T_0 – времени безопасной работы (эксплуатации) ИС.

Также рассмотрен предлагаемый метод, основанный на оценке рисков информационной безопасности, предполагающий суммирование ущерба от происходящих инцидентов и позволяющий оценить время T_0, в течение которого обеспечивается защита ИС.

Модели компьютерных инцидентов и оценки рисков нарушения информационной безопасности

Для описания теоретико-вероятностной модели событий будем полагать, что в случайные моменты времени в системе происходят инциденты I_k, $k = 1, 2, ...$, сопровождаемые ущербом (потерями) и приводящие к нарушению ее штатного режима работы: сбои, компьютерные атаки и тому подобное.

Положим, $u = \{u_1, ..., u_m\}$, $u_i > 0$, $i = 1, 2, ... m$ – возможные варианты значений потерь при выявлении инцидентов. На инциденты реагируют средства защиты информационной системы, и дальнейшее развитие событий может осуществляться по следующим сценариям.

Первый вариант развития сценария состоит в следующем.

При возникновении очередного инцидента I_k величина потерь u_k последовательно сравнивается с максимально допустимой величиной ущерба R_0 . Если

$$u_k < R_0,$$

информационная система продолжает работу в штатном режиме. Иначе необходимо предпринять дополнительные меры защиты и произвести корректировку политики безопасности.

Рассмотрим второй из возможных вариантов сценария.

При выявлении инцидентов $I_{t_1}, I_{t_2}, ..., I_{t_n}, ...$ в моменты времени $t_1, t_2, ..., t_N, ...$ производится последовательное суммирование значений соответствующих ущербов $u_{t_1}, u_{t_2}, ..., u_{t_N}, ...$, вычисление статистики

$$S_N = \sum_{j=1}^{N} u_{t_j}$$

и сравнение ее значения с максимально допустимым значением ущерба R_0.

Если при выявлении инцидента I_{t_n} будет выполнено условие $S < R_0$, в соответствии со сценарием считается, что информационная система работает в штатном режиме. Иначе принимается решение о введении дополнительных мер информационной защиты и пересмотре политики безопасности.

В соответствии с первым сценарием рассмотрим математическую модель реагирования системы на появление инцидентов.

Пусть $\{\xi_t\}$, $t = 1, 2, ..., N$ – последовательность независимых, одинаково распределенных индикаторов,

$$p\{\xi_t = 1\} = p,\ p\{\xi_t = 0\} = 1 - p,$$

соответствующих моментам выявления инцидентов.

Без ограничения общности полагаем, что инциденты появляются независимо друг от друга, а вероятность появления каждого равна p.

Далее, как и ранее, положим, $u = \{u_1,...,u_m\}$ – множество возможных значений величин потерь при наступлении инцидентов и полагаем, что при наступлении инцидента u_k с номером k, вероятность возникновения соответствующего ущерба равна p_k, $\sum_{k=1}^{m} p_k = 1$.

Пусть также для значений $k_1, k_2, ..., k_s, k_j \in \{1, ..., m\}$ имеет место неравенство

$$u_{k_j} \geq R_0, j \in \{1, ..., s\},$$

а для значений $k_j \in \{1, ..., m\} \backslash \{k_1, k_2, ..., k_s\}$ имеет место неравенство

$$u_k < R_0.$$

Положим, далее $q = \sum_{j=1}^{s} p_{k_j}$.

С последовательностью $\{\xi_t\}$ свяжем последовательность индикаторов $\{\zeta_t\}$:

$$\zeta_t = \begin{cases} 1, \textit{если } \xi_t = 1 \textit{ и наступил инцидент } I_k, \textit{ где } k \in \{k_1, k_2, ..., k_s\} \\ 0, \textit{в противном случае} \end{cases}$$

и случайную величину τ:

$$\tau = \min\{t \mid \zeta_t = 1\},$$

момент наступления критического события, при котором информационная система нуждается в дополнительных средствах защиты.

В рамках рассматриваемой модели вероятностное распределение случайной величины τ имеет вид:

$$p\{\tau = k\} = p \cdot q \cdot (1 - p \cdot q)^{k-1}, \quad k = 1, 2, ...,$$

что является геометрическим распределением.

Таким образом, оценкой среднего времени безопасной работы ИС может служить математическое ожидание случайной величины τ равное.

Перейдем ко второму варианту сценария, в котором учитывается общая сумма потерь от инцидентов.

С последовательностью индикаторов выявления инцидентов $\{\xi_t\}$ свяжем последовательность $\{\eta_t\}$:

$$\eta_t = \begin{cases} 1, \textit{если } S_{t-1} < R_0, S_t \geq R_0 \\ 0, \textit{в противном случае} \end{cases}, t = 1, 2, ...,$$

последовательность моментов времени, при которых общая сумма потерь от инцидентов превышает установленную границу R_0. Очевидно, что $\{\eta_t\}$ – это последовательность моментов времени, при которых имеют место критические события. В этих случаях система не обеспечивает требуемый уровень защиты и, следовательно, нуждается в ее усилении.

Определим случайную величину χ:

$$\chi = \min\{t \mid \eta_t = 1\},$$

момент первого наступления критического события.

В рамках введенной теоретико-вероятностной модели для решения поставленной выше задачи найдем распределение случайной величины χ: $p\{\chi = N\}$, где N – натуральное число.

Обозначим $\overline{\alpha}_N = (\alpha_1, ..., \alpha_m)$ – первичную спецификацию исходов последовательности $\{\xi t\}_t^N$, где α_s – число инцидентов, при которых потери составили величину u_s, $s = 1, 2,..., m$, $\alpha_1 + ... + \alpha_m = N$ и положим:

$$\mathrm{M}_{N-1}\left(u_k\right) = \left\{\overline{\alpha}_{N-1} \mid \alpha_1 + ... + \alpha_m = N-1, R_0 - u_k \le \sum_{s=1}^{m} \alpha_s \cdot u_s < R_0\right\}.$$

Тогда для вероятности первого наступления критического события получаем выражение:

$$p\left\{\chi = N\right\} = \sum_{k=1}^{m} p_k \cdot \sum_{\overline{\alpha}_{N-1} \in M_{N-1}(u_k)} \left(1-p\right)^{\alpha_0} \cdot p^{N-\alpha_0} \cdot \prod_{k=1}^{m} p_k^{\alpha_k}\left(\alpha_{N-1}\right),$$

а также

$$p\left\{\chi = N\right\} = \sum_{k=1}^{m} p_k \sum_{\overline{\alpha}_{N-1} \in M_{N-1}(u_k)} C_{N-1}^{m} \cdot \prod_{s=1}^{m} p_s^{\alpha_s}$$

На основании последнего соотношения могут быть построены оценки времени безопасной эксплуатации информационной системы.

Далее в статье представлены результаты экспериментов по построению прогнозных моделей в задачах определения возможной величины потерь юридических лиц от реализации компьютерных инцидентов. Дано описание экспериментов по вычислению времени безопасной эксплуатации информационной системы.

Прогнозная модель потери от хищений со счетов юридических лиц с использованием платежных карт

В настоящее время финансово-кредитным организациям приходится принимать различные меры для противодействия попыткам хищений денежных средств со счетов как юридических, так и физических лиц. С этой целью каждая транзакция проверяется на легитимность путем применения ряда процедур. В случае выявления признаков мошенничества проведение транзакции останавливается для проведения дополнительных процедур подтверждения. Однако в ряде случаев мошенникам все же удается совершить несанкционированные операции, далее называемые неостановленными.

В данном эксперименте прогнозная модель строилась для изучения динамики потери от совершенных мошенниками несанкционированных операций (хищений) со счетов юридических лиц с использованием платежных карт. В табл. 1 приведена статистика объема несанкционированных операций со счетами юридических лиц за период с 1 квартала 2018 года по 4 квартал 2019 года, представленная на официальном сайте Банка России [16]. Данные за указанный период взяты в связи с тем, что в последующие периоды сведения о возникающих потерях организаций полностью перестали публиковаться.

В эксперименте для построения прогнозной аппроксимирующей функции $F(x)$ использовались данные за период с 1 квартала 2018 года по 4 квартал 2018 года, за нулевое значение по оси OX принята дата 4 квартал 2017 года.

Для данного эксперимента аппроксимирующая функция $F(x)$ имела вид:

$$F_1(x) = 510.34 + 26.72\sqrt{x} + 167.25 \cdot \sin\left[2 \cdot x\right] - 6.05 \cdot \cos\left[2 \cdot x\right] + 75.92 \cdot \cos\left[4 \cdot x\right] +$$

$$+ 100.62 \cdot \cos\left[3 \cdot x\right] - 125.08 \cdot \sin\left[x\right] - 58.22 \cdot \cos\left[x\right].$$

График, прогнозной функции $F_1(x)$ представлен на рис. 1.

Таблица 1

Объем несанкционированных операций (хищений) со счетами организаций

Table 1

Volume of unauthorized transactions (embezzlement) with accounts of organizations

Период	Объем несанкционированных операций (хищений, млн руб.)
1 квартал 2018 г.	352,6
2 квартал 2018 г.	342,5
3 квартал 2018 г.	428,7
4 квартал 2018 г.	770,6
1 квартал 2019 г.	405,6
2 квартал 2019 г.	440,3
3 квартал 2019 г.	343,7
4 квартал 2019 г.	380,0

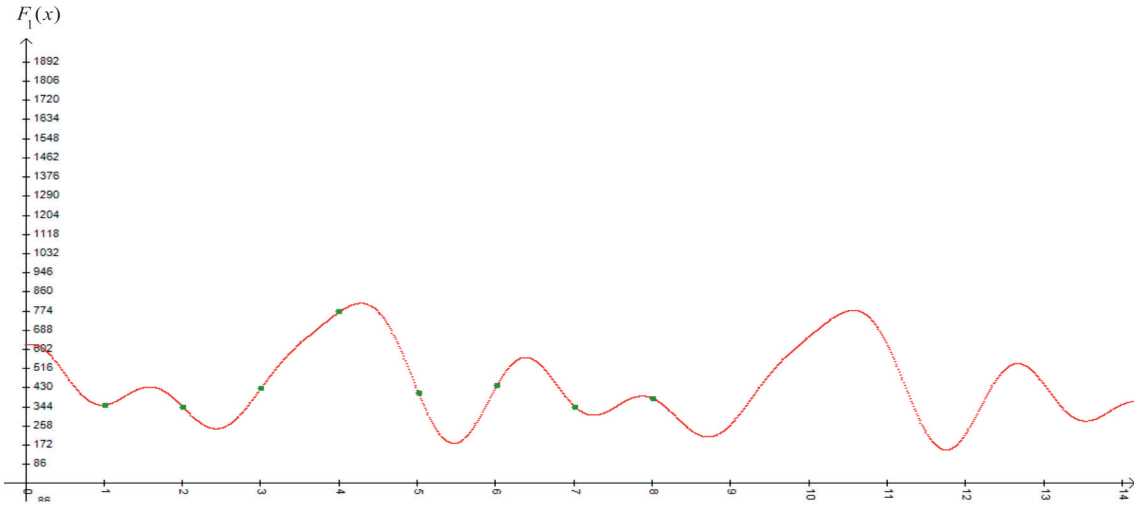


Рис. 1. График функции $y = F_1(x)$

Fig. 1. Graph of the function $y = F_1(x)$

В табл. 2 приведены результаты эксперимента по построению прогнозных значений ущерба от несанкционированных операций (хищений) со счетов юридических лиц с использованием платежных карт на период с 1 квартала 2020 года по 2 квартал 2021 года.

В данном примере, в соответствии с примененным выше методом построения прогнозной модели динамики инцидентов, связанных с объемом несанкционированных операций со счетами юридических лиц с использованием платежных карт, функция их числа $f_1(x)$ в зависимости от времени t имеет вид:

$$f_1(t) = \alpha_1 + \alpha_2\sqrt{t} + \alpha_3 \sin[2{\cdot}t] - \alpha_4 \cos[2{\cdot}t] + \alpha_5 \cos[4{\cdot}t] + \alpha_6 \cos[3{\cdot}t] - \alpha_7 \sin[{\cdot}t] - \alpha_8 \cos[t],$$

где α_i, $i = 1, 2, ..., 8$ – соответствующие константы.

Таблица 2

Прогноз объема несанкционированных операций (хищений)
со счетами организаций

Table 2

Forecast of the volume of unauthorized transactions (embezzlement)
with accounts of organizations

Период	Объем несанкционированных операций (млн руб.)
1 квартал 2020 г.	383
2 квартал 2020 г.	262
3 квартал 2020 г.	657
4 квартал 2020 г.	625
1 квартал 2021 г.	220
2 квартал 2021 г.	446

При $t \geq 0$ для функции $f_1(x)$ имеет место неравенство:

$$f_1(t) \leq \alpha_9 + \alpha_2\sqrt{t},$$

где $\alpha_9 = \alpha_1 + \alpha_3 + ... + \alpha_8$.

Обозначим через N число юридических лиц (организаций, информационных систем), понесших ущерб от действий злоумышленников.

Тогда на основании уравнение (1) прогнозная модель среднего ущерба юридического лица имеет вид:

$$\frac{1}{N} f(t) \leq \frac{1}{N}\left(\alpha_9 + \alpha_2\sqrt{t}\right),$$

при этом оценка для времени T_0 безопасной эксплуатации ИС может быть найдена из уравнения

$$\frac{1}{N}\left(\alpha_9 + \alpha_2\sqrt{t}\right) = R_0,$$

откуда получаем:

$$T_0 \geq \left(\frac{N \cdot R_0 - \alpha_9}{\alpha_2}\right)^2,$$

где R_0 – максимально допустимый ущерб для организации.

В табл. 3 представлен расчет параметра T_0 – времени (в годах) безопасной эксплуатации ИС, вычисления проведены для значений параметра N=10.

Таблица 3

Время (в годах) безопасной эксплуатации информационной системы

Table 3

Time (in years) of safe operation of the information system

R_0 (тыс. руб). / N	1100	1150	1200	1250	1300
10	1,1	4	8,6	15	23

Прогнозная модель неостановленных нелегитимных операций (хищений) со счетов юридических лиц

В рассматриваемом эксперименте прогнозная модель строилась для изучения динамики инцидентов, связанных с количеством неостановленных несанкционированных операций с использованием платежных карт. В табл. 4 приведена статистика в процентном соотношении количества неостановленных нелегитимных операций за период с 1 квартала 2018 года по 4 квартал 2019 года, представленная в [16].

Таблица 4

Процент неостановленных нелегитимных операций

Table 4

Percentage of unstoppable illegitimate operations

Период	Процент неостановленных несанкционированных операций (хищений)
1 квартал 2018 г.	0,27
2 квартал 2018 г.	0,55
3 квартал 2018 г.	0,28
4 квартал 2018 г.	0,61
1 квартал 2019 г.	0,51
2 квартал 2019 г.	0,54
3 квартал 2019 г.	0,51
4 квартал 2019 г.	0,39

В данном эксперименте для построения прогнозной функции $F(x)$ использовались данные за период с 1 квартала 2018 года по 4 квартал 2019 года, за нулевое значение по оси OX принята дата 4 квартал 2016 года.

Для данного эксперимента аппроксимирующая функция $F(x)$ имела вид:

$$F_2(x)=\frac{1.62}{\sqrt{x}}-0.053\cdot\sin x+0.075\cdot\cos x\frac{1.2}{x}+0.075\cdot\cos 3x+5\cdot10^{-4}\cdot x^2-0.039\cdot\sin 4x-0.07\cdot\cos 4x.$$

График прогнозной функции $F_2(x)$ представлен на рис. 2.

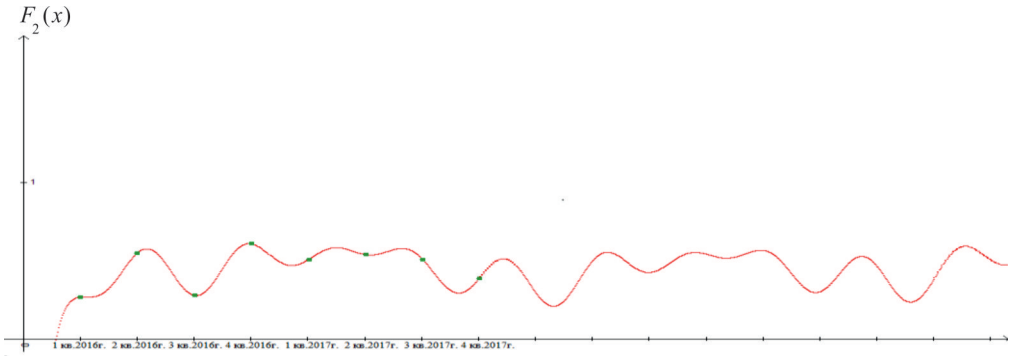


Рис. 2. График функции $y = F_2(x)$

Fig. 2. Graph of the function $y = F_2(x)$

В табл. 5 приведены результаты эксперимента по построению прогнозных значений количества неостановленных несанкционированных операций с использованием платежных карт на период с 1 квартала 2020 года по 2 квартал 2021 года.

Таблица 5

Прогноз количества неостановленных нелегитимных операций

Table 5

Forecast of the number of unidentified illegitimate operations

Период	Процент неостановленных нелегитимных операций
1 квартал 2020 г.	0,3
2 квартал 2020 г.	0,5
3 квартал 2020 г.	0,43
4 квартал 2020 г.	0,54
1 квартал 2021 г.	0,57
2 квартал 2021 г.	0,3
3 квартал 2021 г.	0,47
4 квартал 2021 г.	0,39

В соответствии с примененным выше методом построения прогнозной модели динамики инцидентов, связанных с относительным количеством неостановленных несанкционированных операций с использованием платежных карт, функция их числа имеет вид:

$$f_2(t) = \frac{\alpha_1}{\sqrt{t}} - \alpha_2 \cdot \sin t + \alpha_3 \cdot \cos t - \frac{\alpha_4}{t} + \alpha_5 \cdot \cos 3t + \alpha_6 \cdot t^2 - \alpha_7 \cdot \sin 4t - \alpha_8 \cdot \cos t,$$

где α_i, $i = 1, 2, ..., 8$ – соответствующие константы.

При $t \geq 1$ для функции $f_2(t)$ имеет место неравенство:

$$f_2(t) \leq \alpha_9 + \alpha_6 \cdot t^2,$$

где $\alpha_9 = \alpha_1 + \alpha_2 + \alpha_{3\,+}\alpha_4 + \alpha_5 + \alpha_7 + \alpha_8$.

Обозначим через N количество несанкционированных операций с использованием платежных карт, а через U – средний ущерб, наносимый одной транзакции.

Тогда прогнозная модель среднего ущерба, наносимого информационной системе организации за исследуемый промежуток времени имеет вид:

$$U \cdot N \cdot f(t) \leq U \cdot N \cdot (\alpha_9 + \alpha_6 \cdot t^2).$$

При этом оценка для времени T_0 безопасной эксплуатации ИС может быть найдена как корень уравнения относительно неизвестного аргумента t:

$$U \cdot N \cdot (\alpha_9 + \alpha_6 \cdot t^2) = R_0,$$

откуда получаем:

$$T_0 \geq \sqrt{\left(\frac{R_0/(U \cdot N) - \alpha_9}{\alpha_6}\right)},$$

где R_0 – максимально допустимый ущерб, наносимый организации.

В табл. 6 представлены примеры расчетов времени (в годах) безопасной эксплуатации информационной системы (параметр T_0). Расчеты проведены для значений $N=1000$ нелегитимных операций с целью использования платежных карт.

Таблица 6

Время (в годах) безопасной эксплуатации информационной системы

Table 6

Time (in years) of safe operation of the information system

R_0(млн руб) / U (тыс. руб).	20	30	40	50	60	70	80	90	100
1	0,5	0,62	0,73	0,82	0,91	1	1,06	1,33	1,2

Заключение

В статье исследованы подходы к оценке защищенности информационных систем. Рассмотрены проблемы построения и исследования модели управления рисками информационной безопасности, в которой учитывается накопление ущерба от реализации инцидентов, приводящих к нарушению информационной безопасности.

Построены алгоритмическая и теоретико-вероятностная модели инцидентов, приводящих к нарушению информационной безопасности и предусматривающих различные сценарии работы информационной системы в зависимости от имеющего место ущерба. На основании изложенных результатов построены прогнозные оценки параметра T_0 – времени безопасной работы информационной системы. В качестве примера построены экспериментальные прогнозные модели количества нелегитимных операций со счетами юридических лиц, в том числе с использованием платежных карт. Данные модели основаны на данных о реальных инцидентах и строятся с применением разработанных ранее методов прогнозирования состояния динамических систем путем нахождения непрерывных аппроксимирующих функций. Полученные результаты могут быть применены для дальнейшего развития методологии оценки защищенности информационных систем, в частности, для оценки времени их безопасной работы.

Список литературы

1. Экспертиза и аудит информационной безопасности. [Электронный ресурс]. Режим доступа: https://sudexpa.ru/expertises/ekspertiza-i-audit-informatcionnoi-bezopasnosti/ (дата обращения 17.02.2020)
2. Аудит информационных систем. Регола-мониторинг. [Электронный ресурс]. Режим доступа: https://spb.systematic.ru/about/news/regola-monitoring.htm (дата обращения 20.02.20)
3. Обзор рынка SIEM-систем. [Электронный ресурс]. Режим доступа: https://www.antimalware.ru/node/11 637 (дата обращения 15.03.2020).
4. **Artemyev V., Kostogryzov A., Rudenko J., Kurpatov O., Nistratov G., Nistratov A.** Probabilistic methods of estimating the mean residual time before the next parameters abnormalities for monitored critical systems. Proceedings of the 2nd International Conference on System Reliability and Safety (ICSRS- 2017), December 20–22, 2017, Milan, Italy, pp. 368–373
5. **Kostogryzov A, Nistratov A.** Probabilistic methods of risk predictions and their pragmatic applications in life cycle of complex systems. In “Safety and Reliability of Systems and Processes”, Gdynia Maritime University, 2020. pp. 153–174. DOI: 10.26408/srsp-2020

6. **Kostogryzov A., Nistratov A., Nistratov G.** (2020) Analytical Risks Prediction. Rationale of System Preventive Measures for Solving Quality and Safety Problems. In: Sukhomlin V., Zubareva E. (eds) Modern Information Technology and IT Education. SITITO 2018. Communications in Computer and Information Science, vol 1201. Springer, pp. 352–364.
7. **Егошин Н. С.** Модель типовых угроз безопасности информации, основанная на модели информационных потоков // Доклады Томского государственного университета систем управления и радиоэлектроники. 2021. Т. 24. № 3. С. 21–25.
8. **Кондаков С. Е., Рудь И. С.** Модель процесса проведения компьютерных атак с использованием специальных информационных воздействий // Вопросы кибербезопасности. 2021. № 5(45). С. 12–20.
9. **Калашников А. О., Бугайский К. А., Аникина Е. В.** Модели компьютерных атак //Информация и безопасность. 2019. Т. 22. № 4. С. 517–538.
10. **Ермакова А. Ю.** Разработка методов прогнозирования на примере анализа средств вычислительной техники//Промышленные АСУ и контроллеры. 2017. № 1. С. 28−34.
11. **Ермакова А. Ю., Лось А. Б.** Исследование прогнозных моделей динамической системы на примере прогноза инцидентов информационной безопасности // Компьютерные науки и информационные технологии: сборник статей Международной научной конференции. Саратов: Издательский центр «Наука», 2018 С. 144–149.
12. **Ермакова А. Ю.** Об оценке точности прогнозирования состояний динамической системы методом построения аппроксимирующих функций //Промышленные АСУ и контроллеры. 2018. № 5. С. 36−42.
13. **Ермакова А. Ю.** Об одном подходе к оценке защищенности информационной системы на основе анализа инцидентов // Системы высокой доступности. 2018. № 4. С. 32−35.
14. **Ермакова А. Ю.** Модели DDoS-атак и исследование защищенности информационной системы от данного типа угроз // Промышленные АСУ и контроллеры. 2019. № 12. С. 54−59. DOI: 10.25791/asu.12.2019.1074
15. **Ермакова А. Ю.** Модель компьютерной атаки в условиях ограниченных возможностей защиты и построение прогнозных моделей компьютерных инцидентов // Промышленные АСУ и контроллеры. 2020. № 6. С. 50−57. DOI: 10.25791/asu.6.2020.1194
16. **Калашников А. И.** Обзор несанкционированных переводов денежных средств за 2019 год. Электронный ресурс (режим доступа – свободный): https://ural.ibbank.ru/files/files/aterials 2018/45%20Kalashnikov.pdf/ (дата обращения 11.04.2020).

References

1. Examination and audit of information security. [Electronic resource]. Access mode: https:// sudexpa.ru/expertises/ ekspertiza-i-audit-informatcionnoi-bezopasnosti/ (date of application 17.02.2020).
2. Audit of information systems. Regola-monitoring. [Electronic resource]. Access mode: https:// spb.systematic.ru/about/news/regola-monitoring.htm/ (date of application 20.02.2020).
3. Market overview of SIEM systems. [Eelectronic resource]. Access mode: https://www.antimal ware.ru/node/11 637 (date of application 15.03.2020).
4. **Artemyev V., Kostogryzov A., Rudenko J., Kurpatov O., Nistratov G., Nistratov A.** Probabilistic methods of estimating the mean residual time before the next parameters abnormalities for monitored critical systems. Proceedings of the 2nd International Conference on System Reliability and Safety (ICSRS- 2017), December 20-22, 2017, Milan, Italy, pp. 368-373
5. **Kostogryzov A, Nistratov A.** Probabilistic methods of risk predictions and their pragmatic applications in life cycle of complex systems. In “Safety and Reliability of Systems and Processes”, Gdynia Maritime University, 2020. pp. 153-174. DOI: 10.26408/srsp-2020

6. **Kostogryzov A., Nistratov A., Nistratov G.** (2020) Analytical Risks Prediction. Rationale of System Preventive Measures for Solving Quality and Safety Problems. In: Sukhomlin V., Zubareva E. (eds) Modern Information Technology and IT Education. SITITO 2018. Communications in Computer and Information Science, vol 1201. Springer, pp.352-364.
7. **Egoshin N. S.** Model' tipovy'x ugroz bezopasnosti informacii, osnovannaya na modeli informacionny'x potokov / N. S. Egoshin //Doklady' Tomskogo gosudarstvennogo universiteta sistem upravleniya i radioe'lektroniki. – 2021. – T. 24. – № 3. – S. 21-25.
8. **Kondakov S. E.** Model' processa provedeniya komp'yuterny'x atak s ispol'zovaniem special'ny'x informacionny'x vozdejstvij / S. E.Kondakov, I. S. Rud' // Voprosy' kiberbezopasnosti. – 2021. № 5(45). S. 12-20.
9. **Kalashnikov A. O.** Modeli kolichestvennogo ocenivaniya komp'yuterny'x atak / A.O. Kalashnikov, K.A. Bugajskij, E.V. Anikina //Informaciya i bezopasnost'. – 2019. – T. 22. – № 4. – S. 517-538.
10. **Ermakova A. Y.** Razrabotka metodov prognozirovaniya na primere analiza sredstv vichisliterlnoy techniki//Promishlennie ASU i kontrolleri. –2017. – № 1. – С. 28−34.
11. **Ermakova A.Y., Los A.B.,** Issledovanie prognoznich modeley dinamicheskoj sistemi na primere prognoza insidentov informatinnoj bezopasnosti//Kompjuternie nauki i informationnie technologii: sbornik statei Mezdunarodnoj nauchnoj konferentcii. Saratov: Izdatelskij Center «Nauka», 2018 – С. 144-149.
12. **Ermakova A. Y.** Ob otcenke tochnosti prognozirovanija sostojanij dinamicheskoj sistemi metodom postroenija approksimirujushej funktcii// Promishlennie ASU i kontrolleri. –2018. – № 5. – С.36−42.
13. **Ermakova A.Y.** Ob odnom podchode k otcenke zashishennosti informatcionnoj sistemi na osnove analiza intcidentov//Systemi visokoj dostupnosti. –2018. – № 4. – С. 32−35.
14. **Ermakova A. Y.** Modeli DDoS atak i issledovanie zashishennosti informatcionnoj sistemi ot dannogo tipa ugroz // Promishlennie ASU i kontrolleri.– 2019. – №12. – С. 54−59. DOI: 10.25791/asu.12.2019.1074
15. **Ermakova A. Y.** Model kompjuternoj ataki v uslovijach ogranichennoj vozmojnosti zashiti i postroenie prognoznich modelej compjuternich incidentov // Promishlennie ASU i kontrolleri.–2020. – № 6. – С. 50−57. DOI: 10.25791/asu.6.2020.1194
16. **Kalashnikov A. I.** Review of unauthorized money transfers for 2017. Electronic resource (free access mode) :https://ural.ib-bank.ru/files/files/aterials2018/45%20Kalashnikov.pdf/ (date of application 11.04.2019)

Информация об авторах

А.Ю. Ермакова, доцент кафедры информационного противоборства МИРЭА – Российского технологического университета

А.Б. Лось, доцент кафедры компьютерной безопасности Национального исследовательского университета «Высшая школа экономики»

Information about the Authors

A.Yu. Ermakova, associate professor Department Russian Technological University, Moscow

A.B. Los, associate professor Department National Research University Higher School of Economics

Статья поступила в редакцию 27.02.2023;
одобрена после рецензирования 24.03.2023; принята к публикации 24.03.2023
The article was submitted 27.02.2023;
approved after reviewing 24.03.2023; accepted for publication 24.03.2023

Научная статья

УДК 004.827
DOI 10.25205/1818-7900-2023-21-1-32-45

Программная система визуализации и проверки согласованности оценочных знаний экспертов

Елена Дмитриевна Малаева[1], Гульнара Эркиновна Яхъяева[2]

Новосибирский государственный университет
Новосибирск, Россия

[1]e.malaeva@g.nsu.ru
[2]g.iakhiaeva@g.nsu.ru

Аннотация

Экспертные оценки применяются повсеместно и при решении широкого диапазона задач. При этом зачастую возникает проблема несогласованности множества экспертных оценок. В данной работе предложен алгоритм проверки оценочных экспертных знаний на согласованность. В результате работы алгоритма мы не только получаем ответ о том, являются ли данные согласованными или нет, но также и визуализируем исходные данные в виде дерева. В случае если введенные экспертные оценки не согласованы, на построенном дереве «подсвечиваются» ребра, в которых возникает несогласованность. Алгоритм также предлагает два альтернативных способа разрешения несогласованности оценок. В статье описывается программная система, разработанная на основе данного алгоритма.

Ключевые слова

экспертные оценки, субъективная вероятность, согласованное означивание, нечеткая модель

Для цитирования

Малаева Е. Д., Яхъяева Г. Э. Программная система визуализации и проверки согласованности оценочных знаний экспертов // Вестник НГУ. Серия: Информационные технологии. 2023. Т. 21, № 1. С. 32–45. DOI 10.25205/1818-7900-2023-21-1-32-45

Software System for Visualization and Checking the Consistency of Experts' Evaluative Knowledge

Elena D. Malaeva[1], Gulnara E. Yakhyaeva[2]

Novosibirsk State University

[1]e.malaeva@g.nsu.ru
[2]g.iakhiaeva@g.nsu.ru

Abstract

Expert evaluations are used everywhere and in solving a wide range of problems, but often there is a problem of inconsistency in the set of expert evaluations. In this paper, we propose an algorithm for checking the evaluative expert knowledge for consistency. As a result of the algorithm, we not only get an answer about whether the data is consistent or not, but also visualize the original data in the form of a tree. If the introduced expert evaluations are inconsistent, the constructed tree "highlights" the edges in which there is an inconsistency. The algorithm also offers two alternative ways of resolving evaluation inconsistencies. The article describes a software system developed on the basis of this algorithm.

Keywords
expert evaluations, subjective probability, consistent attribution, fuzzy model

For citation
Malaeva E. D., Yakhyaeva G. E. Software System for Visualization and Checking the Consistency of Experts' Evaluative Knowledge. *Vestnik NSU. Series: Information Technologies,* 2023, vol. 21, no. 1, pp. 32–45. (in Russ.) DOI 10.25205/1818-7900-2023-21-1-32-45

Введение

Экспертное оценивание – это процедура получения оценок (оценки) на базе мнения группы экспертов (эксперта) в каких-либо вопросах с целью дальнейшего принятия наиболее подходящего решения, совершения выбора. Осведомленность, знания, накопленный опыт, способность смотреть наперед, интуиция экспертов побуждают отдельных людей и целые команды обращаться к специалистам за помощью. Самым распространенным способом получения и анализа качественной информации в ситуациях, когда остро чувствуется дефицит объективных данных, названы как раз экспертные оценки [1]. Специалисты со своим, безусловно, внушительным набором полезных способностей и качеств могут находить наилучшие выходы из различных ситуаций, более прибыльные сделки, перспективные и многообещающие направления развития и становления и многое другое, их потенциал неограничен.

Экспертные оценки применяются повсеместно и при решении широкого диапазона задач [2]. Например, при постановке диагноза консилиумом врачей, выборе группы космонавтов из целой группы претендующих на место в ней специалистов, при выборе инвестиционных проектов для реализации среди имеющихся, при одобрении проектов научно-исследовательских работ для спонсорства из огромного числа заявок и так далее.

Методы экспертных оценок при наличии коллектива специалистов можно разделить на две группы [3]: методы коллективной работы экспертной группы и методы получения индивидуального мнения членов экспертной группы. Первая отличается тем, что специалисты в ходе дискуссии должны прийти к общему мнению, которое и представят в виде результата совместной работы. Во второй же группе мы собираем оценки экспертов и анализируем их уже самостоятельно.

На сегодняшний день выделяются следующие методы коллективной работы экспертной группы [4]:

Метод коллективной генерации идей. Это метод, который основан на мотивации творческой работы специалистов с помощью коллективного обсуждения определенной проблемы. При реализации данного метода соблюдается ряд правил: запрет на оценку выдвигаемых идей, лимитирование времени на одного человека, приоритет отдается специалисту, развивающему предшествующую идею, фиксация всех высказанных идей [5].

Метод «635». Шесть экспертов в течение пяти минут должны придумать и записать три идеи, далее лист передается по кругу, и каждый следующий человек записывает свои три идеи за пять минут, базирующиеся на уже имеющихся на листе идеях.

Метод «Дельфи». В процессе эксперты отвечают на вопросы в несколько этапов, при этом после каждого им предоставляются анонимные результаты предыдущего раунда с пояснением причин, по которым был выбран тот или иной ответ. Таким образом, у людей будет шанс переосмыслить свои суждения, именно поэтому считается, что разброс оценок будет уменьшаться, а оценка – стремиться к истинной [6].

Метод «комиссий». Эксперты неоднократно собираются для обсуждения одного и того же вопроса с заранее намеченным планом обсуждения и большим объемом исходной информации.

Метод написания сценария. Экспертами устанавливается последовательность гипотетических событий, связанных друг с другом причинно-следственными связями, поэтому получается модель процесса, а не только модель конечного результата [7].

К основным индивидуальным методам получения мнения относятся следующие [8]:

Метод «интервью». Диалог прогнозиста с экспертом в формате вопрос-ответ, что позволяет получить наиболее честные и точные ответы, благодаря возможности уточнять детали в реальном времени, но количество усилий по сбору и обработке результата сильно превышает количество усилий для других методов.

Аналитический метод. Данный метод заключается в подробном изучении и анализе экспертом решаемой проблемы, но не подходит для оценки сложных ситуаций из-за ограниченности знаний одного эксперта в смежных областях, что могут дополнительно понадобиться для работы [9].

Также существуют математико-статические методы экспертных оценок [10]:

Метод простой ранжировки. Эксперты располагают факторы, признаки или характеристики в порядке собственного предпочтения, далее подсчитывается величина, означающая среднее значение важности факторов (чем меньше величина, тем важнее данный фактор) [11].

Метод весовых коэффициентов. Эксперты оценивают факторы, признаки или характеристики весовыми коэффициентами (есть две вариации: сумма всех должна быть равна фиксированному числу или у самого важного фактора вес фиксирован, а остальным присваиваются коэффициенты, равные долям этого числа).

Метод последовательных сравнений. При этом методе происходит систематическая проверка оценок с помощью их последовательного сравнения. Данный метод является наиболее точным из представленных, но запрашивает большое количество экспертов и усилий в сравнении с двумя предыдущими методами [12].

Метод парных сравнений. Эксперты выбирают в каждой паре наиболее значимый фактор, признак или наиболее значимую характеристику, когда их много или когда все объекты равнозначны. Таким образом, проводится статистически обоснованный анализ экспертных оценок. Данный метод реализуем тяжелее, чем метод ранжирования, но проще, чем метод последовательных сравнений.

В дополнение ко всем уже перечисленным недостаткам вышеуказанных методов в работе [3] также выделены сложность формирования группового мнения и вероятность давления авторитетов. В данной работе предлагается технология, позволяющая описанные выше проблемы. Она позволяет экспертам высказываться не в строго заявленном формате и не только по какому-то конкретному вопросу, находит несогласованность в их мнениях, а также по всем этим оценкам рассчитывает новые, что облегчает формирование группового мнения и исключает возможность давления авторитетов.

1. Математические основы программной системы

Понятие «субъективной вероятности» было введено в 30-х годах прошлого века Фрэнком Рамсеем и подразумевает степень уверенности эксперта в наступлении того или иного события. Ее применяют тогда, когда невозможно воспользоваться объективной вероятностью по причине неполноты или отсутствия данных о наблюдениях в прошлом, из-за высокой стоимости получения объективной вероятности [13].

В работах [14, 15] рассматривается теоретико-модельная формализация субъективной вероятности через аппарат нечетких моделей.

Определение 1. *Тройку* $\mathfrak{A}_\mu = \langle A, \sigma, \mu \rangle$ *будем называть* ***нечеткой моделью***, *если*

1. *Отображение* $\mu: S(\sigma_A) \to [0,1]$ *является нечеткой, счетно-аддитивной мерой;*
2. $\varphi \sim \psi \Rightarrow \mu(\varphi) = \mu(\psi)$, *для любых* $\varphi, \psi \in S(\sigma_A)$.

Задавая нечеткую модель $\mathfrak{A}_\mu$ как формализацию предметной области, мы должны располагать знанием о субъективной вероятности всех предложений множества $S(\sigma_A)$. Однако эксперт (или даже группа экспертов) может не обладать таким полным знанием. Более того, множество $S(\sigma_A)$ может оказаться бесконечным. Тем не менее, существует конечное множество

событий предметной области S, о вероятностных значениях которых эксперт может дать свою субъективную оценку.

Таким образом, на входе мы имеем конечное множество предложений $S \in S(\sigma_A)$ и означивание η_S: $S \to [0,1]$. Далее встает необходимость решения следующих задач.

Задача 1. Проверить является ли означивание η_S корректным (т. е. согласованным).

Задача 2. Если оценка некорректна, найти причины возникновения некорректности и предложить эксперту пересмотреть часть своих означиваний.

Задача 3. Если оценка корректна, то для произвольного предложения $\varphi \in S(\sigma_A)\backslash S$ найти всевозможные значения истинности, совместные с данным означиванием.

В работе [16] Задача 3 была частично решена, было доказано, что для любого предложения $\varphi \in S(\sigma_A)\backslash S$ множество всевозможных значений истинности, согласованных с данным означиванием, образует интервал. В данной работе мы сосредоточимся на решении первых двух задач.

Определение 2. *Рассмотрим множество предложений S и отображение η_S: $S \to [0,1]$. Означивание η_S* ***согласуется с*** *нечеткой моделью $\mathfrak{A}_\mu = \langle A, \sigma, \mu \rangle$, если для любого предложения $\varphi \in S$ выполняется равенство $\eta_S(\varphi) = \mu(\varphi)$.*

Означивание назовем ***согласованным****, если существует нечеткая модель сигнатуры σ_S, с которой данное означивание согласуется.*

Замечание 1. *Пусть $\mathfrak{A}_\mu = \langle A, \sigma, \mu \rangle$ – нечеткая модель. Тогда для любого $\varphi \in S(\sigma_A)$ имеем*

$$\mu(\varphi) = \mu(\psi_1) + \cdots + \mu(\psi_n),$$

где $\psi_1 \vee \ldots \vee \psi_n$ – СДНФ предложения φ.

Доказательство следует из свойств семантической эквивалентности формул.

Рассмотрим конечное множество бесквантворных предложений $S = \{\varphi_1, \ldots, \varphi_n\}$. Введем следующие обозначения:

$S_\alpha(\sigma_s) = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ – множество атомарных предложений сигнатуры σ_s;

$S_\beta(\sigma_S) = \{\omega_1, \ldots, \omega_{2^n}\} = \{\bigwedge_{i=1}^{n} \varepsilon_i \alpha_i \mid \varepsilon_i \in \{\emptyset, \neg\}, \alpha, \in S_\alpha(\sigma)\}$ – множество всех максимальных конъюнктов сигнатуры σ.

Тогда каждому предложению $\varphi_i \in S$ можно подставить в соответствие некоторый двоичный вектор $\alpha_i = (\alpha_{i1}, \ldots, \alpha_{i2^n})$, где $\alpha_{ij} = 1$ тогда и только тогда, когда конъюнкт $\omega_j \in S_\beta(\sigma_s)$ входит в СДНФ предложения φ_i.

Таким образом, множеству предложений S поставим в соответствие двоичную матрицу

$$\mathbb{S} = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} \alpha_{11} & \cdots & \alpha_{12^n} \\ \cdots & \cdots & \cdots \\ \alpha_{n1} & \cdots & \alpha_{n2^n} \end{pmatrix}.$$

Теорема 1. Пусть $S = \{\varphi_1, \ldots, \varphi_n\}$ – *конечное множество бесквантворных предложений, $\mathbb{S}$ – соответствующая этому множеству двоичная матрица и η_S: $S \to [0,1]$. Отображение η_S согласованно тогда и только тогда, когда система линейных уравнений*

$$\mathbb{S} \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_{2^n} \end{pmatrix} = \begin{pmatrix} \eta_S(\varphi_1) \\ \vdots \\ \eta_S(\varphi_n) \end{pmatrix} \tag{1}$$

совместна и имеет решения с ограничениями

$$\begin{cases} x_i \in [0,1], \text{для любого } i = \overline{1, 2^n}; \\ \sum_{i=1}^{2^n} x_i = 1. \end{cases} \tag{2}$$

Доказательство. Обозначим через C_s множество констант сигнатуры σ_s. Так как S – множество бесквантворных предложений, то $C_s \neq \emptyset$.

Пусть означивание η_S согласовано. Тогда найдется такая нечеткая модель $\mathfrak{A}_\mu = \langle C_S, \sigma_S, \mu \rangle$, определенная на множестве констант C_s, что для любого $\phi \in S$ имеем $\eta_S(\varphi) = \mu(\varphi)$. Следовательно, по Замечанию 1 кортеж истинностных значений $(\mu(\omega_1), ..., \mu(\omega_2{}^n))$ является решением системы (1) с ограничениями (2).

Допустим теперь, что кортеж $(a_1, ..., a_2{}^n)$ является решением системы (1) с ограничениями (2). Выполнение ограничений позволяет рассматривать отображение $\omega_i \mapsto a_i$ как вероятностную меру, определенную на σ-алгебре $S_\beta(\sigma_s)$, $S(\sigma_s)$. Полученное таким образом вероятностное пространство будет эквивалентно некоторой нечеткой модели $\mathfrak{A}_\mu = \langle C_S, \sigma_S, \mu \rangle$.

Теорема доказана.

2. Архитектура программной системы

Для проверки оценочных экспертных знаний на согласованность была разработана программная система, которая состоит из трех модулей:

1. Модуль парсинга формул.
2. Модуль проверки согласованности оценок и визуализация.
3. Модуль исправлений несогласованностей.

Опишем принципы работы каждого из перечисленных модулей.

2.1. Модуль парсинга формул

Формулы на вход системы передаются в виде строковых выражений. Парсинг переданных формул преобразует каждое из выражений в дерево объектов.

Парсинг выражения выполняется за один проход по переданной строке за счет алгоритма, схожего с алгоритмом Дейкстра. Алгоритм работает с двумя стеками: первый хранит операции, второй – формулы. Принцип алгоритма заключается в следующем:

- Проходим исходную строку (рис. 1).

```
@Override
public Formula parseFormula(String formulaString) {

    char[] formulaCharArray = formulaString.replace(" ", "").toCharArray();

    int currentIndex = 0;
    while (currentIndex < formulaCharArray.length) {
```

Рис. 1.
Fig. 1.

- При нахождении **предиката** заносим его в стек формул.
- При нахождении **оператора** заносим его в стек операций (рис. 2).

```
private void processOperation(OperationStackItem operationStackItem) {

    switch (operationStackItem.getStackItemType()) {

        case OPERATION -> handleOperation(operationStackItem);

        case OPENING_BRACKET -> handleOpeningBracket(operationStackItem);

        case CLOSING_BRACKET -> handleClosingBracket(operationStackItem);

    }

}
```

Рис. 2.
Fig. 2.

- Из стека операций выталкиваем все операции с приоритетом выше.
- Аргументы выталкиваемой операции заполняем элементами из стека формул.
- Если элементов из стека формул не хватает для заполнения всех аргументов операции, выражение некорректное.
- При нахождении открывающейся скобки, заносим ее в стек операций.
- При нахождении закрывающей скобки, выталкиваем из стека операций все операторы до открывающейся скобки, а открывающуюся скобку удаляем из стека. При этом выполняем те же действия, что и при выталкивании операций с более высоким приоритетом.

2.2. Модуль проверки согласованности оценок и визуализация

Для проверки согласованности оценочных знаний нам необходимо привести все введенные формулы к виду СДНФ. Для этого выделяются все атомарные предложения, входящие в заданный набор формул, и фиксируются в лексикографическом порядке. После этого строится таблица истинности для каждого предложения. По таблице истинности строится система линейных уравнений

$$\begin{cases} \sum_{j=1}^{m} \alpha_{ij} x_j = \mu_i, \qquad i = \overline{1,n}; \\ \sum_{j=1}^{m} x_j = 1. \end{cases}$$

где n – количество формул, введенных пользователем, и m – количество всевозможных дизъюнктов, содержащих все атомарные предложения, взятые с отрицанием или без. Коэффициент α_{ij} равен 1, если конъюнкт под номером j содержится в полном СДНФ представлении предложения под номером i, и 0 в противном случае. Коэффициент μ_i является оценочным значением i-го предложения. Очевидно, что $0 \le \mu_i \le 1$. Переменная x_j является искомым оценочным значением j-го конъюнкта. Согласно Теореме 1, множество предложений, введенное пользователем, согласованно тогда и только тогда, когда данная система имеет решения при ограничениях $0 \le x_j \le 1$ для любого $j = \overline{1,m}$.

По теореме Кронекера–Капелли [17] проверяется совместность системы. При решении системы уравнений может возникнуть три ситуации:

Случай 1. Система имеет решения при заданных ограничениях.

Случай 2. Система не имеет решений при заданных ограничениях, но имеет решения, выходящие за пределы ограничений.

Случай 3. Система не имеет решений.

В первом случае множество предложений согласованно, и в нем нет конфликтов. В качестве примера этого случая возьмем входные данные из табл. 1.

Таблица 1

Входные данные для случая согласованного множества предложений

Table 1

Input Data for the Case of a Consistent Set of Proposals

$\varphi_1 = P(a)$ v ! $Q(b)$,	0,8
$\varphi_2 = (P(a)$ V $Q(b))$ & $(P(a)$ V ! $Q(b))$,	0,7

Приводим формулы φ_1 и φ_2 к виду СДНФ:

$$\varphi_1 = \big(P(a) \vee Q(b)\big) \vee \big(P(a) \vee !\,Q(b)\big) \vee \big(!\,P(a) \vee !\,Q(b)\big)$$
$$\varphi_2 = \big(P(a) \vee Q(b)\big) \vee \big(P(a) \vee !\,Q(b)\big)$$

и строим систему линейных уравнений:

$$\begin{cases} x_1 + x_2 + x_3 = 0{,}8; \\ x_1 + x_2 = 0{,}7; \\ x_1 + x_2 + x_3 + x_4 = 1. \end{cases}$$

Очевидно, что эта система имеет решение с ограничениями $0 \le x_j \le 1$ для любого $j = \overline{1,4}$.

Для визуализации данных из табл. 1 производится отрисовка графа, где в качестве листьев представлены конъюнкты из СДНФ формул, которые постепенно собираются в сами формулы, что мы подали, с помощью дизъюнкции между собой (в конечном итоге превращаясь в дизъюнкцию всех конъюнкций с вероятностью 1), а ребра означают отношение частичного порядка между ними. У каждой вершины есть свое оценочное значение (это может быть интервал). По умолчанию вершине присваивается оценочное значение в виде интервала [0; 1], далее эти значения могут корректироваться в зависимости от введенных пользователем значений. Граф для примера из табл. 1 представлен на рис. 3.

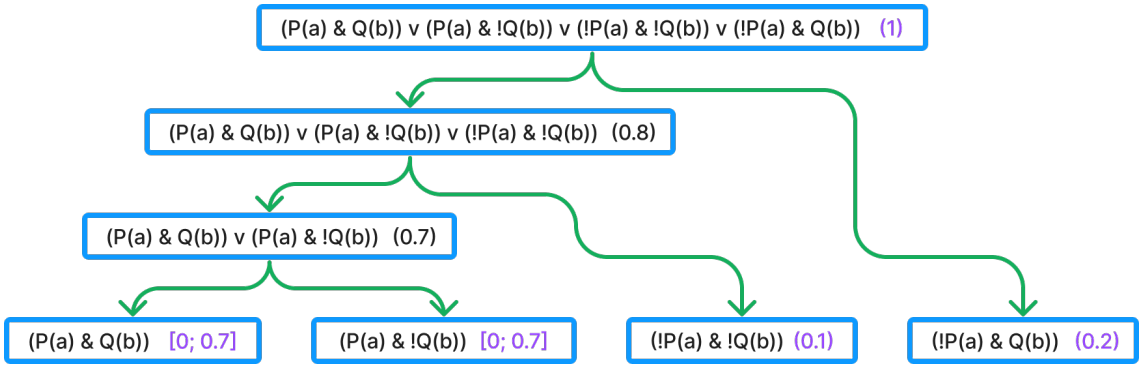


Рис. 3. Граф для примера из табл. 1
Fig. 3. Graph for an example from the Table 1

Рассмотрим теперь ситуацию, когда система линейных уравнений не имеет решений при заданных ограничениях, но имеет решения, выходящие за пределы ограничений. В этом случае с помощью построенного графа можно также увидеть конкретные места, где возникают проблемы т. е. где именно происходит рассогласование. В качестве примера такой ситуации рассмотрим входные данные из табл. 2.

Таблица 2

Входные данные для случая несогласованного множества предложений

Table 2

Input Data for the Case of an Inconsistent Set of Proposals

$\varphi_1 = P(a) \vee !\,Q(b)$,,	0,65
$\varphi_2 = (P(a) \vee Q(b)) \,\&\, (P(a) \vee !\,Q(b))$,,	0,8

Этим входным данным соответствует следующая система линейных уравнений:

$$\begin{cases} x_1 + x_2 + x_3 = 0{,}65; \\ x_1 + x_2 = 0{,}8; \\ x_1 + x_2 + x_3 + x_4 = 1. \end{cases}$$

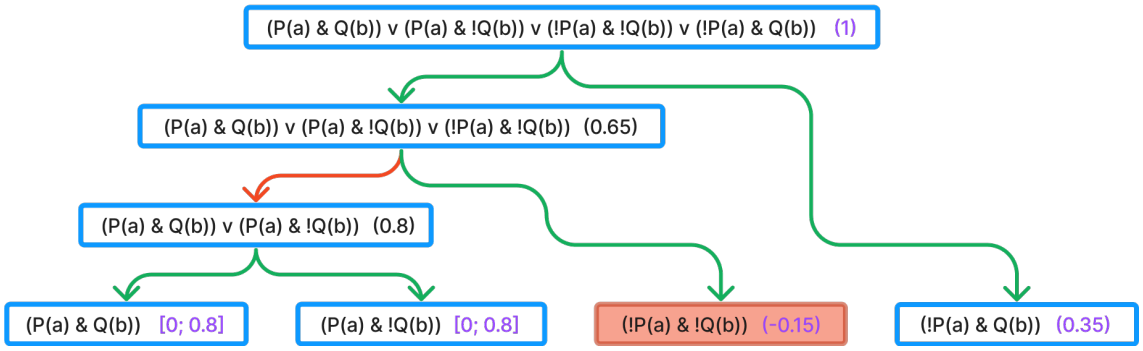


Рис. 4. Граф для примера из табл. 2
Fig. 4. Graph for an example from the Table 2

Очевидно, что любое решение этой системы выходит за рамки ограничений, так как значение конъюнкта $!P(a) \vee !Q(b)$ принимает значение –0,15. Это видно и на построенном дереве (рис. 4).

Далее происходит проверка всех ребер построенного дерева. Вес вершин на пути от листа до корня дерева должен монотонно, нестрого возрастать. В том месте, где это правило нарушается, и происходит рассогласование оценок. На рис. 4 это ребро окрашено в красный цвет. Во всех остальных переходах монотонность возрастания оценок соблюдена, поэтому ребра окрашены в зеленый цвет.

Случай, когда система линейных уравнений несовместна, является одним из вариантов несогласованности оценок экспертов, в этом случае конкретное место с некорректностью найти невозможно из-за отсутствия в принципе решений, поэтому граф не строится. Пользователю в этом случае предлагается пересмотреть набор критериев (т. е. множество формул), которые необходимо оценивать.

2.3. Модуль исправлений несогласованностей

Если все решения построенной системы линейных уравнений не удовлетворяют заданным ограничениям, «проблемные» места видны на визуализации. При некоторой корректировке оценочных значений данные могут быть согласованы. В разработанной программной системе предлагается два подхода корректировки оценочных значений:

– подход ближайшего верного значения;
– подход равномерного распределения нагрузки [18].

В первом случае предлагается заменить значение «верхнего» или «нижнего» конъюнкта до ближайшего верного. Пример такой замены можно увидеть на рис. 5. Здесь мы либо заменяем значение конъюнкта над некорректным ребром до значения «нижнего» конъюнкта, либо заменяем значение конъюнкта под некорректным ребром до значения «верхнего» конъюнкта. Таким образом, отношение становится верным, а значит возникший конфликт разрешается.

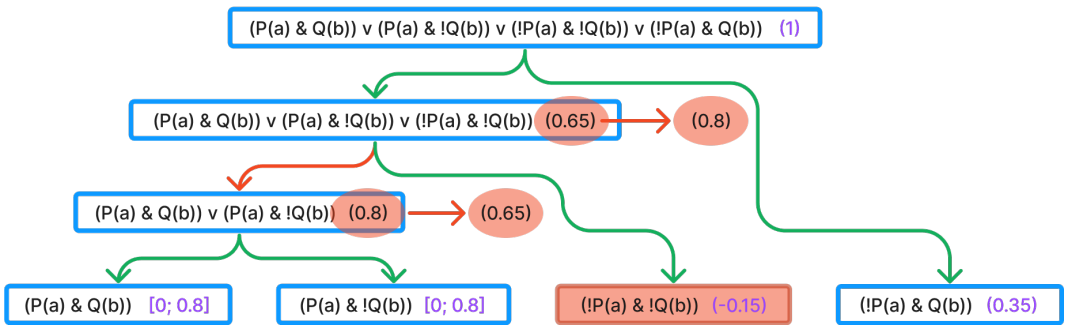


Рис. 5. Пример использования подхода ближайшего верного значения
Fig. 5. An example of using the nearest-right-value approach

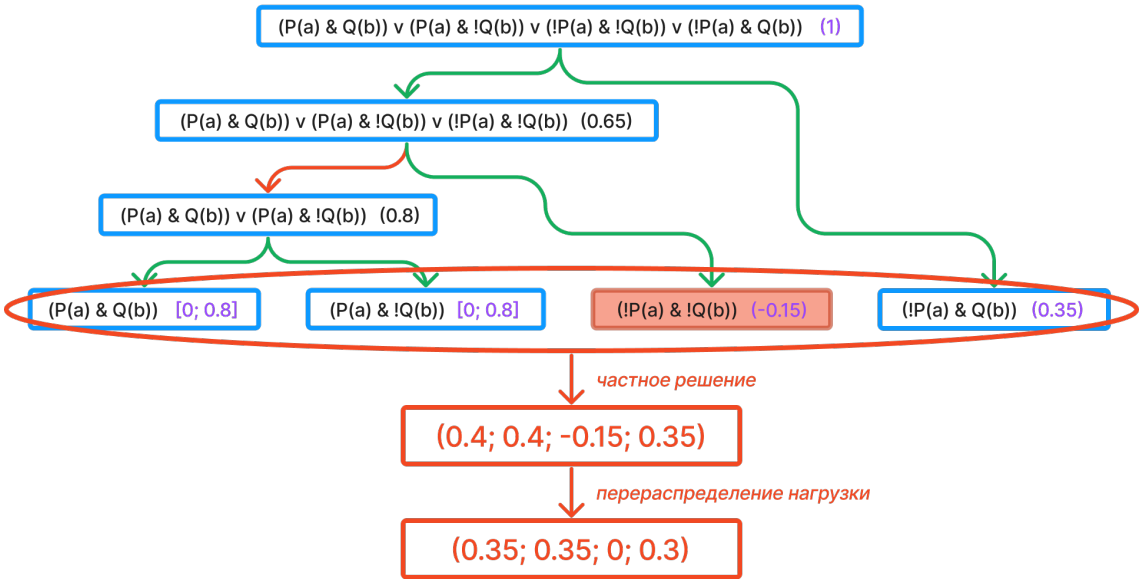


Рис. 6. Пример работы алгоритма равномерного распределения нагрузки
Fig. 6. An example of the work of the algorithm for uniform load distribution

При втором подходе сначала берется одно частное решение системы ‹a_1, ..., a_m› и считается его суммарное нижнее отклонение:

$$\beta = \sum_{j=1}^{m} min\{0, a_j\}.$$

Затем отклонение β равномерно распределяется на положительные значения вектора ‹a_1, ..., a_m›, т. е. строится вектор ‹b_1, ..., b_m›, где

$$b_j = \begin{cases} 0, & a_j \le 0; \\ a_j + \dfrac{\beta}{k}, & a_j > 0. \end{cases}$$

На следующем шаге алгоритма меняем вектор введенных экспертных оценок ‹μ_1, ..., μ_n› так, чтобы вектор ‹b_1, ..., b_m› являлся решением системы уравнений. Пример работы алгоритма равномерного распределения нагрузки показан на рис. 6.

3. Интерфейс программной системы

Интерфейс программной системы был реализован с использованием библиотеки JavaFX. Реализовано разделение Model, View, Controller за счет поддержки FXML и возможности писать контроллеры, привязывая его методы к элементам UI.

Пользователь, используя диалоговое окно, может задавать события, описывая их в виде формул логики предикатов, используя операции конъюнкции, дизъюнкции, импликации и отрицания, а также кванторы существования и всеобщности, и субъективные вероятности наступления этих событий в виде чисел от 0 до 1. Пользователь может ввести любое число формул и их оценочные значения (рис. 7).

После нажатия кнопки «Построить граф» строится визуализация введенных данных. Узлы, в которых представлены программно-вычисленные оценки, отрисованы голубым цветом, а узлы, в которых представлены оценки, заданные пользователем, – фиолетовым. Скриншот работы программной системы в случае, если введенные пользователем оценки согласованны, представлен на рис. 8.

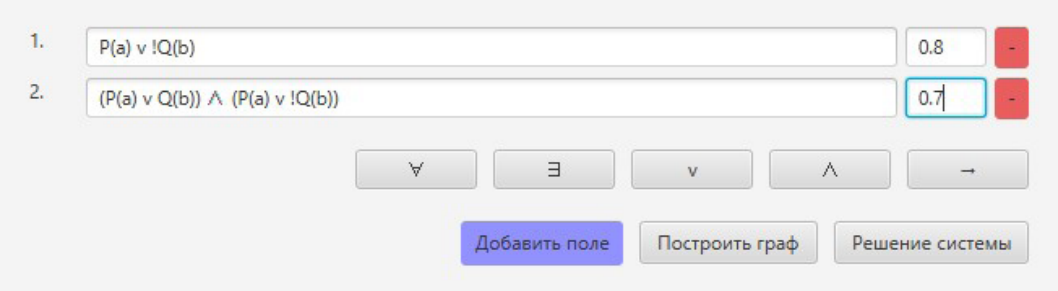


Рис. 7. Диалоговое окно
Fig. 7. Dialog window

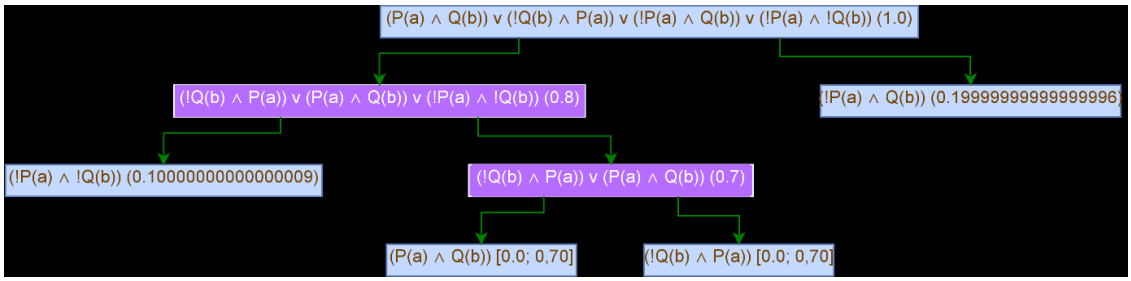


Рис. 8. Результат работы программной системы в случае согласованных оценок
Fig. 8. The result of the work of the software system in the case of consistent evaluations

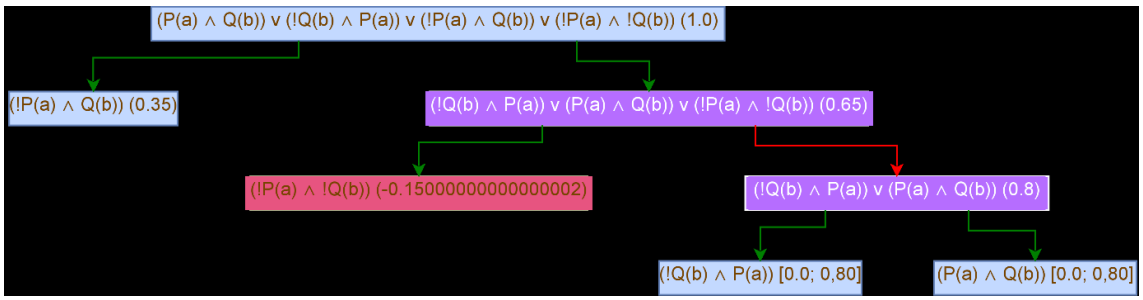


Рис. 9. Результат работы программной системы в случае несогласованных оценок
Fig. 9. The result of the work of the software system in the case of inconsistent evaluations

Если же для введенных пользователем данных не существует решения с заданными ограничениями, то места несогласованности подсвечиваются цветом. Розовые узлы – это узлы с отрицательной вероятностью. Красные ребра – это ребра, на которых нарушается свойство монотонности оценок. Скриншот работы программной системы в случае, если введенные пользователем оценки не согласованны, представлен на рис. 9.

Пользователь может изменить введенные ранее оценки формул и снова проверить их согласованность. В этом случае, в качестве подсказки, появляются дополнительные два окна с вариантами исправления несогласованностей: методом ближайшего верного значения (рис. 10) и методом равномерного распределения нагрузки (рис. 11).

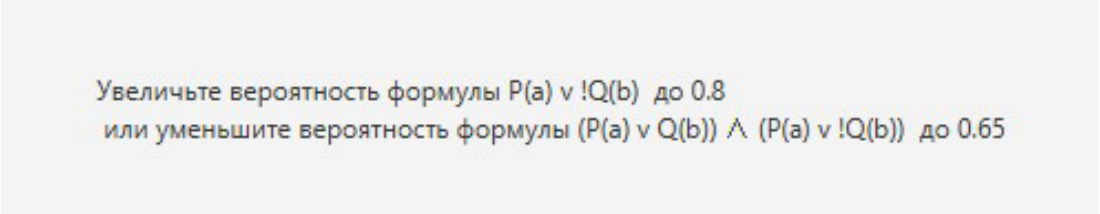


Рис. 10. Результат работы метода ближайшего верного значения
Fig. 10. The result of the work of the nearest-right-value approach

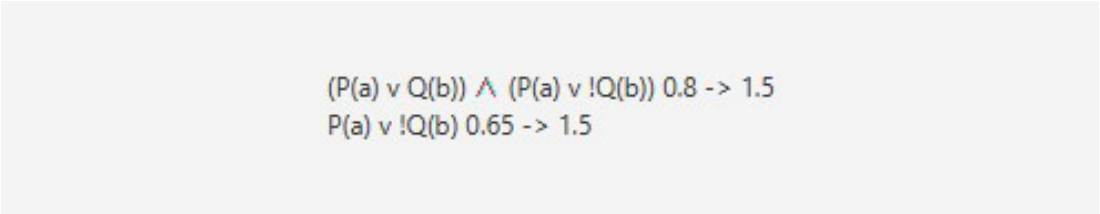


Рис. 11. Результат работы метода равномерного распределения нагрузки
Fig. 11. The result of the work of the method for uniform load distribution

Заключение

В данной работе описана программная система проверки согласованности множества субъективных оценок событий в той или иной предметной области. Описание событий формализуется на языке логики предикатов, оценки этих событий являются действительными числами из интервала [0,1]. Для проверки согласованности оценочных знаний строится система линейных уравнений, которая однозначно определяет, есть проблема или нет, но наглядно не может показать, где именно возникла несогласованность. Для визуализации «узких мест» по введенным пользователем данным строится граф и предлагаются варианты разрешения конфликтов оценок.

На данный момент в разработанной программной системе реализуется глобальная проверка согласованности и отрисовка единого графа для всего множества введенных пользователем формул (формальных описаний событий). Этот подход имеет минус «комбинаторного взрыва», заключающийся в том, что при увеличении числа понятий предметной области (т. е. при увеличении сигнатуры допустимых формул) рост количества переменных в системе линейных уравнений происходит экспоненциально. Для решения этой проблемы мы планируем внедрить в систему иерархическую кластеризацию введенных пользователем формул, что позволит внедрить локальную проверку согласованности оценочных знаний.

Список литературы

1. **Гуцыкова С.** Метод экспертных оценок. Теория и практика. М.: Институт психологии РАН, 2011. 144 с.
2. **Palchunov D. E., Tishkovsky D. E., Tishkovskaya S. V., Yakhyaeva G. E.** Combining logical and statistical rule reasoning and verification for medical applications. *2017 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON),* Novosibirsk, Russia, 2017, p. 309-313. DOI: 10.1109/SIBIRCON.2017.8109895.
3. **Данелян Т. Я.** Формальные методы экспертных оценок // Статистика и экономика. 2015. № 1. С. 183-187. DOI: 10.21686/2500-3925-2015-1-183-187.

4. **Бурцева Т. А., Зуева И. А.** Направления совершенствования методик мониторинга реализации стратегий развития регионов в условиях цифровой информационной среды // Вестник Московского университета имени С. Ю. Витте. Серия 1: Экономика и управление. 2018. Т. 27, № 4. С. 43-50. DOI: 10.21777/2587-554x-2018-4-43-50.
5. **Волковская В. М., Захаров В. В.** Метод коллективной генерации идей // Инновационные процессы в сфере информационных технологий и современного образования в регионах России: сб. научн. ст. по материалам Всероссийской научно-практической конференции, Ставрополь, 16–17 ноября 2020 года / Ставрополь: АГРУС, 2020. С. 85–89.
6. **Стончюте К. Э., Гурбо А. А., Пузыревская А. А.** Методы экспертных оценок: Метод Дельфи // Научное знание современности. 2021. Т. 53, № 5. С. 16–19.
7. **Датенко В. И.** Применение ориентированных графов при прогнозировании исходов в многокомпонентных задачах // Прикладные информационные системы в технологиях наземного транспорта (машиностроение): материалы Всероссийской научно-практической конференции с международным участием, Таганрог, 20–21 декабря 2018 года / Таганрог: ЭльДирект, 2019. С. 47–50.
8. **Евстигнеева Е. О., Новикова И. В.** Метод экспертных оценок в прогнозировании // Проблемы и перспективы развития экспериментальной науки: сб. ст. Международной научно-практической конференции, Новосибирск, 28 ноября 2019 года / Уфа: OMEGA SCIENCE, 2019. С. 72–74.
9. **Цугунян А. М., Кваско М. А.** Методы принятия стратегических решений на микро- и макроуровнях // Современная мировая экономика: проблемы и перспективы в эпоху развития цифровых технологий и биотехнологии: сб. научн. ст. Международной научной конференции, Москва, 29–31 марта 2019 года / М.: КОНВЕРТ, 2019. С. 53–55.
10. **Козенко И. А.** Использование экспертных оценок при определении потребительских предпочтений // Актуальные вопросы современной экономики. 2018. № 9. С. 287–296.
11. **Демина Л. М., Дивина Т. В.** Исследование потребительских предпочтений на основе экспертных оценок : учеб.-методич. пособие. М.: МГИУ, 2012. 56 с.
12. **Дивина Т. В., Петракова Е. А., Вишневский М. С.** Основные методы анализа экспертных оценок // Экономика и бизнес: теория и практика. 2019. № 7. С. 42–44. DOI: 10.24411/2411-0450-2019-11072
13. **Дулесов А. С., Семенова М. Ю.** Субъективная вероятность в определении меры неопределенности состояния объекта // Фундаментальные исследования. 2012. № 3. С. 81–86.
14. **Яхъяева Г. Э., Пальчунова О. Д.** Нечеткие модели как формализация оценочных знаний экспертов // Двадцатая национальная конференция по искусственному интеллекту с международным участием, КИИ-2022: Труды конференции. В 2-х т., Москва, 21–23 декабря 2022 года / М.: Национальный исследовательский университет «МЭИ», 2022. С. 97–109.
15. **Yakhyaeva G.** Method for Verifying the Logical Correctness of Experts' Evaluative Knowledge. 2022 IEEE International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), Yekaterinburg, Russian Federation, 2022, p. 850–854.
16. **Пальчунов Д. Е., Яхъяева Г. Э.** Нечеткие алгебраические системы // Вестник НГУ. Серия: Математика, механика, информатика. 2010. Т. 10, № 3. С. 76–93.
17. **Ильин В. А., Ким Г. Д.** Линейная алгебра и аналитическая геометрия. М.: Проспект, 2007. 400 с.
18. **Yakhyaeva G., Skokova V.** Subjective Expert Evaluations in the Model-Theoretic Representation of Object Domain Knowledge. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2021, p. 152–165.

References

1. **Gutsykova S.** Method of expert evaluations. Theory and practice. Moscow: Institute of Psychology RAS, 2011. (in Russ.)
2. **Palchunov D. E., Tishkovsky D. E., Tishkovskaya S. V., Yakhyaeva G. E.** Combining logical and statistical rule reasoning and verification for medical applications // *2017 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*, Novosibirsk, Russia, 2017. Pp. 309–313. DOI 10.1109/SIBIRCON.2017.8109895
3. **Danelyan T. Y.** Formal methods of expert estimations // Statistics and Economics. 2015. No. 1. Pp. 183–187 (in Russ.) DOI 10.21686/2500-3925-2015-1-183-187
4. **Burtseva T. A., Zueva I. A.** Directions of perfection of methods of monitoring the implementation of development strategies of regions in terms of digital information // Vestnik Moskovskogo universiteta imeni S. Yu. Vitte. Seriya 1: Ekonomika i upravlenie. 2018. Vol. 27, no. 4. Pp. 43–50. (in Russ.) DOI 10.21777/2587-554x-2018-4-43-50
5. **Volkovskaya V. M., Zakharov V. V.** Method of collective idea generation // Innovatsionnye protsessy v sfere informatsionnykh tekhnologii i sovremennogo obrazovaniya v regionakh Rossii: sbornik nauchnykh statei po materialam Vserossiiskoi nauchno-prakticheskoi konferentsii. Stavropol, November 16–17, 2020. Pp. 85–89. (in Russ.)
6. **Stonchyute K. E., Gurbo A. A., Puzyrevskaya A. A.** Methods of expert evaluations: Delphi method // Scientific knowledge of modernity. 2021. Vol. 53, no. 5. Pp. 16–19. (in Russ.)
7. **Datenko V. I.** Oriented graphs application for the prediction of the outcomes in multicomponent task // Prikladnye informatsionnye sistemy v tekhnologiyakh nazemnogo transporta (mashinostroenie): materialy Vserossiiskoi nauchno-prakticheskoi konferentsii s mezhdunarodnym uchastiem, Taganrog, December 20–21, 2018. Pp. 47–50. (in Russ.)
8. **Evstigneeva E. O., Novikova I. V.** The method of expert evaluations in forecasting // Problemy i perspektivy razvitiya eksperimental'noi nauki: sbornik statei Mezhdunarodnoi nauchno-prakticheskoi konferentsii, Novosibirsk, November 28, 2019. Pp. 72–74. (in Russ.)
9. **Tsugunyan A. M., Kvasko M. A.** Methods for making strategic decisions at the micro and macro levels // Sovremennaya mirovaya ekonomika: problemy i perspektivy v epokhu razvitiya tsifrovykh tekhnologii i biotekhnologii : sbornik nauchnykh statei mezhdunarodnoi nauchnoi konferentsii, Moscow, March 29–31, 2019. Pp. 53–55. (in Russ.)
10. **Kozenko I. A.** The use of expert evaluations in determining consumer preferences // Actual issues of the modern economy. 2018. No. 9. Pp. 287–296. (in Russ.)
11. **Demina L. M., Divina T. V.** Research of consumer preferences based on expert evaluations. Mocow: MSIU, 2012. (in Russ.)
12. **Divina T. V., Petrakova E. A., Vishnevsky M. S.** Basic methods of analysis of expert assessments // Economy and business: theory and practice. 2019. No. 7. Pp. 42–44. (in Russ.) DOI 10.24411/2411-0450-2019-11072
13. **Dulesov A. S., Semyonova M. Y.** Subject probability in measure detection of object state uncertainty // Fundamental'nye issledovaniya. 2012. No. 3. Pp. 81–86. (in Russ.)
14. **Yakhyaeva G. E., Palchunova O. D.** Fuzzy models as a formalization of evaluative knowledge // Dvadtsataya Natsional'naya konferentsiya po iskusstvennomu intellektu s mezhdunarodnym uchastiem, KII-2022 : Trudy konferentsii. Moscow, December 21-23, 2022. Pp. 97–109. (in Russ.)
15. **Yakhyaeva G.** Method for Verifying the Logical Correctness of Experts' Evaluative Knowledge. 2022 IEEE International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), Yekaterinburg, Russian Federation, 2022. Pp. 850–854.
16. **Palchunov D. E., Yakhyaeva G. E.** Fuzzy algebraic systems // Vestnik NSU. Series: Mathematics, mechanics, computer science. 2010. Vol. 10, no. 3. Pp. 76–93. (in Russ.)

17. **Il'in V. A., Kim G. D.** Linear Algebra and Analytic Geometry. Moscow: Prospekt, 2007. (in Russ.)
18. **Yakhyaeva G., Skokova V.** Subjective Expert Evaluations in the Model-Theoretic Representation of Object Domain Knowledge. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2021. Pp. 152–165.

Информация об авторах

Малаева Елена Дмитриевна, студент НГУ

Яхъяева Гульнара Эркиновна, кандидат физико-математических наук, доцент кафедры общей информатики НГУ

Information about the Authors

Elena D. Malaeva, student, Novosibirsk State University (Novosibirsk, Russian Federation)

Gulnara E. Yakhyaeva, Candidate of Physical and Mathematical Sciences, Associate Professor of the Department of General Informatics, Novosibirsk State University (Novosibirsk, Russian Federation)

Статья поступила в редакцию 02.03.2023;
одобрена после рецензирования 18.03.2023; принята к публикации 18.03.2023
The article was submitted 02.03.2023;
approved after reviewing 18.03.2023; accepted for publication 18.03.2023

Научная статья

УДК 004.8
DOI 10.25205/1818-7900-2023-21-1-46-61

Прозрачное уменьшение размерности с помощью генетического алгоритма

Никита Андреевич Радеев

Новосибирский государственный университет
Новосибирск, Россия

n.radeev@g.nsu.ru, orcid.org/0000-0002-4334-5725

Аннотация

Существуют предметные области, где все преобразования данных должны быть прозрачными и объяснимыми (например, медицина и финансы). Уменьшение размерности данных является важной частью предварительной обработки данных, но алгоритмы для него в настоящее время не являются прозрачными. В данной работе мы предлагаем генетический алгоритм для прозрачного уменьшения размерности числовых табличных данных. Алгоритм строит признаки в виде деревьев выражений на основе подмножества числовых признаков из исходных данных и обычных арифметических операций. Он спроектирован так, чтобы стремиться к достижению максимального качества в задачах бинарной классификации и генерировать признаки, объяснимые человеком, что достигается за счет использования в построении признаков операций, понятных человеку. Кроме того, преобразованные алгоритмом данные могут быть использованы в визуальном анализе, если уменьшить размерность до двух. В алгоритме используется многокритериальная динамическая фитнес-функция, предназначенная для построения признаков с высоким разнообразием.

Ключевые слова

генетический алгоритм, уменьшение размерности, построение признаков, интерпретируемость, объяснимый ИИ, символьная регрессия

Для цитирования

Радеев Н. А. Прозрачное уменьшение размерности с помощью генетического алгоритма // Вестник НГУ. Серия: Информационные технологии. 2023. Т. 21, № 1. С. 46–61. DOI 10.25205/1818-7900-2023-21-1-46-61

Transparent Reduction of Dimension with Genetic Algorithm

Nikita A. Radeev

Novosibirsk State University
Novosibirsk, Russian Federation

n.radeev@g.nsu.ru, orcid.org/0000-0002-4334-5725

Abstract

There are domain areas where all transformations of data must be transparent and interpretable (medicine and finance for example). Dimension reduction is an important part of a preprocessing pipeline but algorithms for it are not transparent at the current time. In this work, we provide a genetic algorithm for transparent dimension reduction of numerical data. The algorithm constructs features in a form of expression trees based on a subset of numerical features from the source data and common arithmetical operations. It is designed to maximize quality in binary classification tasks and generate features explainable by a human which achieves by using human-interpretable operations in a feature construction.

Also, data transformed by the algorithm can be used in a visual analysis. The multicriterial dynamic fitness function is provided to build features with high diversity.

Keywords

genetic algorithm, dimension reduction, feature construction, interpretability, explainable AI, symbolic regression

For citation

Radeev N. A. Transparent Reduction of Dimension with Genetic Algorithm. *Vestnik NSU. Series: Information Technologies*, 2023, vol. 21, no. 1, pp. 46–61. (in Russ.) DOI 10.25205/1818-7900-2023-21-1-46-61

Введение

Уменьшение размерности – одна из основных частей процесса анализа данных. Низкоразмерные данные могут быть эффективно визуализированы. Такие визуализации полезны в исследовательском анализе данных для изучения их природы. Меньшим количеством признаков удобно манипулировать и анализировать человеку. Сокращение размерности полезно при обработке больших данных, поскольку оно уменьшает объем данных с минимально возможной потерей информации и позволяет ускорить обработку больших данных [1].

Очень важно, чтобы признаки, генерируемые алгоритмами уменьшения размерности, были интерпретируемыми в таких областях, как медицина, финансы, управление персоналом, правосудие, образование и маркетинг. Другими словами, все области, где решения могут повлиять на человека не лучшим образом с опасностью для здоровья, жизни, финансов, нуждаются в интерпретируемых манипуляциях с данными [2, 3]. Эксперт должен уметь объяснить значение генерируемых признаков в терминологии предметной области. В противном случае решение, предоставляемое алгоритмом, не может быть использовано в производстве, поскольку отсутствие объяснимости приводит к непредвиденным ошибкам, которые недопустимы в таких предметных областях.

Проблему уменьшения размерности можно представить как поиск низкоразмерного представления исходного пространства. Эта идея лежит в основе Manifold Learning [4, 5], которая предполагает, что полезные данные содержатся в низкоразмерном многообразии, вложенном в высокоразмерное пространство. Таким образом, цель состоит в том, чтобы найти представление, которое сохраняет полезную информацию и не содержит бесполезной информации. Полезность информации может быть определена многими способами. Самый прямой способ – это экспертная оценка информации. В то же время это самый сложный способ, потому что он требует наличия эксперта, которого трудно найти. Кроме того, время специалиста стоит дорого. Также существует естественное ограничение на объем информации, который может оценить человек. Таким образом, экспертная оценка подходит в отдельных случаях данных с небольшим количеством признаков. В случае с автоматической оценкой полезности информации существует широкий спектр вариантов. Это может быть качество предсказания модели машинного обучения, линейная разделимость [6] кластеров, принадлежащих разным классам, мера расстояния между такими кластерами или какая-либо пользовательская бизнес-метрика, которая может быть рассчитана на данных.

Генетическое программирование [7–9] – это подход к решению задач оптимизации. У него есть несколько важных свойств, которые отличают этот тип оптимизации от других. В генетической оптимизации существует функция, называемая фитнес, которую необходимо оптимизировать. Требования к фитнес-функции гораздо мягче, чем, например, к функции, оптимизируемой при градиентном спуске [10]. Фитнес-функция может быть недифференцируемой, это самое интересное свойство генетической оптимизации, позволяющее легко создавать сложные фитнес-функции, совершенно не заботясь об их дифференцируемости. Существуют модификации генетической оптимизации, которые не требуют даже точного значения фитнес-функции. Таким алгоритмам нужны только ранги, т. е. отношения «больше-меньше» между фит-

нес-функциями индивидов, для сравнения их друг с другом. В этом случае лучшее решение имеет самый высокий (или самый низкий) ранг, но точное значение функции при этом может быть неизвестно.

Чтобы построить решение, необходимо выбрать такое представление признаков, которое может эффективно мутировать. Символическая (или символьная) регрессия [11, 12] – это подход к нахождению арифметического выражения, которое описывает закон, по которому производятся данные. Он может быть представлен в виде дерева выражений, где лист – это терминал (признак из исходных данных), а узел – операция над дочерними поддеревьями. Деревья могут быть закодированы различными способами: графы [13], обратная польская нотация [14], или менее известные методы, такие как генные выражения [15, 16], которые и используются в данной работе и будут рассмотрены далее.

В этой работе мы предлагаем прозрачный алгоритм уменьшения размеров под названием ГУРу (Генетическое Уменьшение Размерности). ГУРу является эволюционным алгоритмом и реализует общий конвейер генетического алгоритма. Он строит признаки, стремясь обеспечить линейную разделимость пространства признаков и максимизировать среднее расстояние между объектами различных классов.

Обзор существующих работ

Классические подходы строят непрозрачные решения, потому что они оперируют данными с позиции статистики и не используют никаких знаний о предметной области. Например, PCA [17] находит подпространство в исходном пространстве, в котором вдоль осей у данных наибольшая дисперсия. Для аналитика данных является сложной задачей объяснить значение таких признаков с точки зрения предметной области. Наиболее распространенным случаем решения задачи сокращения размерности является использование классических подходов, таких как PCA, когда признаков очень много и работать с таким большим количеством признаков просто невозможно, или t-SNE [18], когда необходимо сделать некоторые интуитивные выводы о распределении данных путем визуализации пространства признаков. С другой стороны, существует несколько работ, посвященных прозрачному сокращению размерности [19–22]. Они используют эволюционный подход для уменьшения размерности данных и получения интерпретируемых человеком признаков. Эти алгоритмы генерируют признаки в виде деревьев выражений, которые имеют тот же уровень интерпретируемости, что и исходные признаки, поскольку операции, применяемые к данным, легко интерпретировать. Поэтому, если имеется большой объем данных, который необходимо уменьшить без потери смысла, эти алгоритмы можно использовать вместо классических подходов, которые не могут уменьшить размерность данных без потери интерпретируемости и генерируют признаки, которые трудно объяснить с помощью терминологии предметной области.

Алгоритм GP-DR

В работе [20] авторы исследуют, насколько хорошо работает простой генетический алгоритм без настройки гиперпараметров (размер популяции и количество эпох) в задаче сокращения размерности. В статье авторы не дают ему названия, поэтому в данной работе назовем этот алгоритм GP-DR (Genetic Programming for Dimension Reduction). Существует несколько фитнес-функций: расстояние между объектами (А), сохранение ранга (на основе расстояния) (Б), функция, оценивающая качество дистилляции кодирующей части автоэнкодера (В), и качество дистилляции всего автоэнкодера (Г). Во всех случаях это алгоритм обучения без учителя.

Алгоритм (А) сравнивает расстояния между объектом в исходном высокоразмерном пространстве и его проекцией в сгенерированном низкоразмерном пространстве. Сумма таких расстояний должна быть минимизирована. В этом заключается суть фитнес-функции, основанной на расстоянии.

Фитнес-функция, сохраняющая ранг (Б), предназначена для сохранения порядка рангов расстояний. Для каждого объекта все остальные объекты могут быть отсортированы по расстоянию и ранжированы. В новом пространстве эти ранги должны быть такими же, как в исходном пространстве. Эта фитнес-функция предназначена для того, чтобы генетический алгоритм минимизировал количество неправильно упорядоченных пар объектов в низкоразмерном пространстве. Объекты с более высокими рангами более важны, чем объекты с более низкими рангами. Таким образом, в фитнес-функции используется взвешивание объектов в соответствии с их рангами.

Другая фитнес-функция (В) использует выходной сигнал кодирующей части автоэнкодера и предназначена для того, чтобы генетический алгоритм генерировал такой же выходной сигнал. Ее можно представить как задачу символьной регрессии на выходе кодировщика и исходном пространстве признаков в качестве цели.

Последняя фитнес-функция (Г) реализует тот же подход, но выход автоэнкодера в целом должен сравниваться с выходом генетического алгоритма. Деревья генетических алгоритмов делятся на деревья кодировщиков и деревья декодировщиков. При этом только кодирующие деревья используются для генерации низкоразмерного пространства после завершения подгонки алгоритма.

Алгоритм GP-MaL-MO

Этот генетический алгоритм основан на идее, что соседи в новом низкоразмерном пространстве должны иметь порядок, аналогичный порядку в исходном высокоразмерном пространстве. Этот алгоритм называется GP-MaL-MO (Genetic Programming for Manifold Learning using a Multi-objective Approach) [21], является расширением алгоритма GP-MaL [23]. Данный алгоритм использует многокритериальную функцию пригодности для построения фронта Парето из решений. Он использует многоцелевой эволюционный алгоритм на основе декомпозиции (MOEA/D) [24] для генерации решений. Алгоритм также генерирует решения в виде деревьев выражений с исходными признаками в листьях и операциями между ними в узлах дерева. Таким образом, GP-MaL-MO генерирует деревья выражений, которые могут быть интерпретированы человеком и могут быть использованы как прозрачный алгоритм уменьшения размерности.

Существует компромисс между качеством и количеством признаков (измерений). Таким образом, пользователь может выбрать решение, удовлетворяющее его требованиям. Алгоритм имеет операторы кроссинговера и мутации, разработанные для того, чтобы GP-MaL-MO мог генерировать решения как с одним деревом, так и со ста деревьями, что приводит к богатому фронту Парето на выходе алгоритма. Как показывают авторы, алгоритм работает так же хорошо, как и классические алгоритмы уменьшения размерности, такие как PCA, LLE и др. [25], MDS [26] и UMAP [27], и полученные с его помощью решения имеют сопоставимое качество.

Описание алгоритма ГУРУ

В ГУРУ реализовано несколько отличительных особенностей:

- Хромосома – это дерево вычислений, которое кодируется методом «генных выражений» [15, 16].
- Фитнес-функция является динамической и изменяется для каждого конструируемого признака.
- Конвейер алгоритма разделен на две части: фаза исследования и фаза эксплуатации.

Все эволюционные алгоритмы имеют схожую конструкцию. Это цикл, начинающийся с оценки стартовой популяции, выбора лучших особей, мутации, кроссинговера особей и создания новой популяции из их детей. Эта процедура повторяется, пока не срабатывают крите-

Таблица 1

Операторы, используемые в ГУРУ

Table 1

Operators Used in GURU

Операция	+	-	*	/	exp	ln	cos	модуль	среднее	max
Количество операндов	2	2	2	2	1	1	1	1	2	2

рии остановки. Выбор генетических операций влияет на стиль поиска алгоритма в пространстве поиска, его скорость и баланс между поведением разведки и эксплуатации.

Далее приводится подробное описание генетических операций и операндов в ГУРУ.

ГУРУ работает с индивидами, которые представляют собой деревья вычислений. Эти деревья содержат терминальные признаки в листьях и операции в узлах. Таким образом, если это дерево будет вычисляться, то на входе будет несколько терминалов, а на выходе – одна функция.

Алгоритм работает с числовыми данными, представленными столбцами входного датафрейма данных. Текущая реализация работает с датафреймами известной в среде аналитиков данных библиотеки pandas. Таким образом, столбцы исходного набора данных, содержащие числовые данные, являются терминальными признаками для алгоритма.

Популяция в ГУРУ – это массив определенного размера, содержащий индивидов. Ее размер меняется между фазами разведки и эксплуатации, но является константой для всех итераций в одной фазе.

В версии ГУРУ, представленной в данной работе, используются обычные арифметические операции из табл. 1. Они могут быть легко интерпретированы аналитиком (при условии, что в конкретной предметной области эти операции между признаками имеют смысл).

В алгоритме используются три типа мутации: инверсия, транспонирование последовательности вставки (IS), транспонирование корневой последовательности вставки (RIS).

- Инвертирующая мутация случайным образом выбирает подпоследовательность в массиве head и инвертирует ее.
- Мутация IS Transpose случайным образом выбирает подпоследовательность внутри головки и вставляет ее в другую позицию, кроме начальной.
- Мутация RIS Transpose делает то же самое, что и IS Transpose. Единственное отличие заключается в том, что RIS Transpose вставляет подпоследовательности только из корневой позиции.

В ГУРУ используются стратегии одноточечного и двухточечного кроссинговера.

- Одноточечный кроссинговер берет два гена, случайным образом выбирает точку и создает двух потомков, объединяя части родителей относительно выбранной точки.
- Двухточечный кроссинговер похож на одноточечный, но в нем выбираются две точки, и потомки строятся путем обмена материалами родителей между этими двумя точками.

В ГУРУ лучшие индивиды отбираются с помощью турнирного отбора. В нем случайным образом выбирается фиксированное число особей и сравниваются их показатели приспособленности. Лучшая особь из каждого турнира переходит в следующее поколение.

ГУРУ оценивает новые признаки по отношению к ранее созданным. Это заставляет алгоритм находить такие новые признаки, которые содержат информацию, не содержащуюся в ранее созданных признаках. Это приводит к построению выразительных компактных признаков. Стоит отметить, что каждый новый признак сам по себе все менее и менее важен, чем ранее сгенерированный, поскольку каждый новый признак зависит от всех предыдущих. Поэтому возможно, что какой-то признак высокого ранга (сгенерированный на поздней эпохе) не содер-

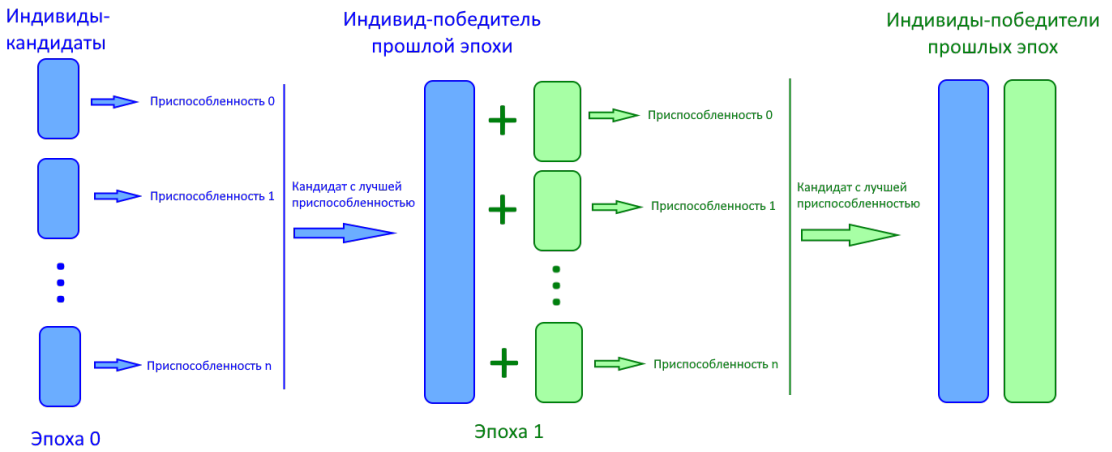


Рис. 1. Конвейер ГУРУ, генерирующий два признака
Fig. 1. GURU pipeline generating two features

жит никакой полезной информации для классификации, но содержит информацию об ошибках всех предыдущих признаков. Это может привести к чрезмерной подгонке (переобучению).

Первый признак оценивается с помощью подгонки к нему функции пригодности и 3-кратной кросс-валидации. Каждый следующий признак оценивается с помощью ранее созданных признаков. Например, второй признак оценивается с помощью оценивания функции пригодности на данных, которые содержат первый признак, полученный из предыдущей эпохи. Это позволяет ГУРУ генерировать признаки таким образом, что каждый новый признак исправляет недостатки всех ранее сгенерированных признаков. Данный процесс проиллюстрирован на рис. 1.

Программирование выражений генов – это разновидность генетического программирования. Отличительной особенностью генных выражений является линейное представление фиксированной длины. Такое представление может быть преобразовано в вычислительное дерево. Хромосома в генном выражении – это строка фиксированной длины. Она делится на головку и хвост, как показано на рис. 2. Голова – это список функций и терминалов постоянной длины. Хвост содержит только терминалы. Его длина зависит от того, сколько операндов требуется головным функциям. Листья дерева – это терминалы, а узлы – функции. Преимуществом генных выражений является простота реализации генетических операций (мутация, кроссинговер, оценка). Недостатком может быть недостаточная выразительность выражений из-за фиксированной длины, что с другой стороны также сокращает пространство поиска решений.

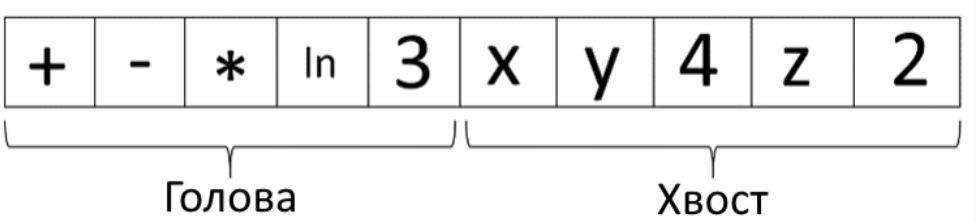


Рис. 2. Генное выражение
Fig. 2. Gene expression

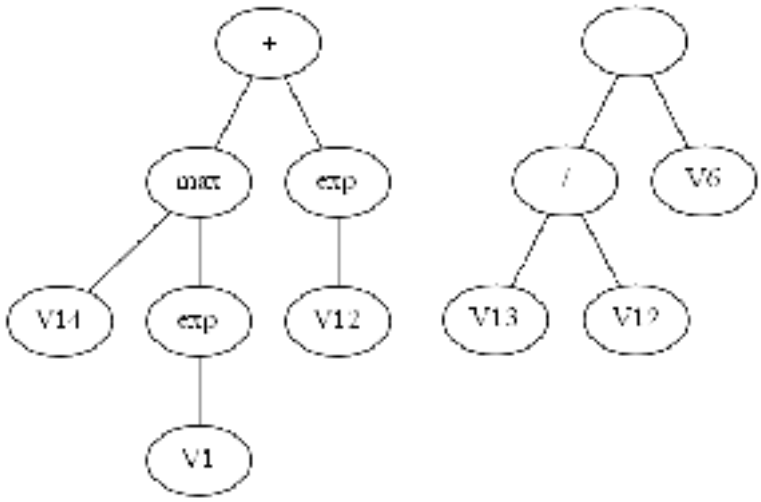


Рис. 3. Дерево, сгенерированное ГУРУ для набора данных «банковский маркетинг»
Fig. 3. Expression tree generated by GURU on “bank-marketing” dataset

Выражение гена однозначно отображается в дерево вычислений, как показано на рис. 3. Однако одно дерево может отображаться на более чем одно генное выражение, поскольку существуют генные выражения, содержащие неиспользуемые операции и терминалы.

Генные выражения – это один из многих методов кодирования дерева вычислений. Он прост в использовании и управлении: регулируя длину генного выражения, мы можем избежать чрезмерной подгонки и регулировать выразительность решения.

Функция пригодности оценивает качество особи. Это одна из самых важных частей генетического алгоритма, поскольку она определяет пространство поиска. Очень важно сделать фитнес-функцию, которая описывает решения с разных точек зрения, чтобы уменьшить вероятность перебора. В данной работе мы используем динамическую многокритериальную фитнес-функцию.

Многокритериальная фитнес-функция использует более одного критерия для оценки особи. В данной работе мы используем два критерия для каждой итерации генерации. Фитнес-функция представлена взвешенной суммой, как в (1). Один из критериев называется примесью, а другой – базовой фитнес-функцией. Базовая функция не изменяется в процессе работы алгоритма и является одной и той же для всех итераций алгоритма. Примесь изменяется на каждой итерации.

$$fitness = \frac{k_1 * fit_{\text{база}} + k_2 * fit_{\text{примесь}}}{k_1 + k_2} \quad (1)$$

В ГУРУ мы используем такую конфигурацию мультикритериальной фитнес-функции:

- Базовая функция – линейная машина опорных векторов [28].
- Примесь – это один из трех алгоритмов: логистическая регрессия, дерево решений или мера расстояния.

Интуиция, лежащая в основе такой мультикритериальной функции, заключается в том, чтобы построить различные признаки благодаря динамической примеси. И в то же время признаки должны строить линейно разделяемое пространство благодаря статической базовой функции, которая является линейной моделью. Такая комбинация приводит к стремлению алгоритмом создать линейно разделимое пространство даже при наличии нелинейной функции-примеси.

Фитнес-функция является динамической, поскольку алгоритм оценивает каждый новый признак с помощью другой фитнес-функции. Пример сокращения до двух измерений показан на рис. 1. Это делает пространство признаков результата более разнообразным и выразительным по сравнению со статической фитнес-функцией, когда существует одна и та же модель для оценки всех генерируемых признаков. В данной работе используются простые фитнес-функции из разных семейств алгоритмов:

- *Логистическая регрессия*, которая должна заставить алгоритм построить признак, делающтй пространство более линейно разделяемым.
- *Мера расстояния*, приводящая к построению признака, который смещает объекты разных классов на большее расстояние. Мы используем среднее попарное расстояние между объектами классов. Оно масштабируется по формуле (2) таким образом, что значение около 0,0 является наихудшим, а 1,0 – наилучшим:

$$distance_{\ scaled} = \frac{1 - e^{-distance}}{1 + e^{-distance}} \tag{2}$$

- *Дерево решений.* Это простая и быстрая модель, принцип работы которой сильно отличается от логистической регрессии и меры расстояния. Лучше использовать признаки, полученные с помощью нелинейной модели, чем с помощью другой линейной модели.

Компромисс между разведкой и эксплуатацией – известная дилемма в машинном обучении. Исследование – это такое поведение алгоритма, когда он совершает большие хаотические скачки в пространстве поиска и потенциально имеет возможность выскочить из локального экстремума и попасть в другой. Такое поведение увеличивает шансы найти глобальный экстремум. Тем не менее, разведка является плохой стратегией для поиска точной точки экстремума. Напротив, в случае поведения эксплуатации алгоритм делает небольшие шаги в пространстве поиска. Это помогает найти точную точку экстремума. Однако эксплуатирующий алгоритм, скорее всего, не найдет ни глобального, ни даже другого локального экстремума, а будет медленно и верно сходиться к тому экстремуму, возле которого находится.

Генетические алгоритмы также страдают от компромисса между разведкой и эксплуатацией. Популяция может содержать похожие особи, которые могут иметь малые вероятности мутации и кроссинговера. Алгоритм с такой конфигурацией будет реализовывать стратегию эксплуатации, поскольку особи меняются медленно. Поэтому алгоритм делает небольшие шаги в пространстве поиска. В других случаях вероятности мутации и кроссинговера достаточно высоки. Тогда алгоритм будет генерировать различные особи в популяции. Это поведение разведки, поскольку каждая особь имеет небольшой шанс выжить и дать потомство с похожими свойствами. Возможно, она мутирует и потеряет свои полезные свойства или потеряет их при скрещивании с другой особью.

В данной работе мы разделили процесс работы алгоритма на две фазы: разведка и эксплуатация. Идея заключается в том, чтобы дать алгоритму свободу в начале процесса поиска генерировать огромное количество различных особей. Они будут отсортированы по значению фитнес-функции, и алгоритм возьмет подмножество лучших особей. Затем алгоритм начнет фазу эксплуатации.

Сбор данных

Бенчмарк (набор данных для оценки качества работы алгоритмов) содержит 12 датасетов, загруженных с сайта проекта OpenML [29]: bank marketing [30], blood transfusion [31], breast cancer wisconsin [32], [33], credit-g [33], diabetes [33], hyperplane [29], ionosphere [34], madelon [35], sonar [36], bioresponse [29], christine [29], guillermo [29]. Была протестирована только задача бинарной классификации. Как видно из табл. 2, все наборы данных в целом имеют числовые признаки и только два набора данных имеют категориальные признаки, чтобы посмо-

треть, как алгоритмы будут работать на наборах данных, где важны категориальные признаки. Для проверки устойчивости алгоритмов есть несколько достаточно больших наборов данных, таких как hyperplane и guillermo. Баланс классов также различается, но нет наборов данных с огромным дисбалансом.

В наборе данных christine имеется 38 бинарных и унарных признаков. В данной работе мы используем их как числовые признаки, где False – 0.0, а True – 1.0.

В табл. 2 в колонке OpenML id представлены идентификаторы наборов данных на сайте.

Таблица 2

Набор данных, используемые в бенчмарке

Table 2

Datasets Used in the Benchmark

Название наборов данных	**OpenML id**	**#признаки числовые / категориальные**	**#образцы**	**Баланс классов**
bank marketing	1461	7/9	45211	7,5 = 39922/5289
blood transfusion	1464	4/0	748	3,2 = 570/178
breast cancer wisconsin	1510	30/0	569	1,7 = 357/212
credit-g	31	7/13	1000	2,3 = 700/300
diabetes	37	8/0	768	1,8 = 500/268
hyperplane	43122	10/0	500000	1,0 = 250000/250000
ionosphere	59	34/0	351	1,8 = 225/126
madelon	1485	500/0	2600	1,0 = 1300/1300
sonar	40	60/0	208	1,1 = 111/97
bioresponse	4134	1776/0	3751	1,2 = 2034/1717
christine	41142	1599/37	5418	1,0 = 2709/2709
guillermo	41159	4296/0	20000	1,5 = 11997/8003

Эксперимент

В этом разделе представлено сравнение с другими прозрачными алгоритмами уменьшения размерности. Все эксперименты запускались десять раз с различными случайными семенами из фиксированного набора. В бенчмарке наборы данных разбиваются на обучающий и тестовый наборы. Затем каждый алгоритм обучается на обучающем множестве. Настроенная модель преобразует как обучающий, так и тестовый наборы. Затем модель классификации обучается на преобразованном обучающем множестве и оценивается на преобразованном тестовом множестве. Используется метрика AUC-ROC. Все десять результатов, рассчитанных для каждого алгоритма на каждом наборе данных, усредняются, и этот средний балл используется для сравнения алгоритмов.

Конфигурация ГУРУ была одинаковой во всех бенчмарках. Она представлена в табл. 3. Гиперпараметры вероятности кроссинговера и мутации были предварительно настроены на бенчмарке. Все остальные гиперпараметры были определены интуитивно. Таким образом, возможно, что ГУРУ может достичь лучшего качества в другой конфигурации, что является предметом отдельного исследования.

Таблица 3

Конфигурация ГУРУ

Table 3

Configuration of GURU

Параметр	**Значение**
случайные состояния	[72, 73, 74, ..., 81]
режим слияния	категории
выходные признаки	2
поколения	21
население	1600
режим работы функций терминала	все
вероятность мутации	0,25
вероятность кроссинговера	0,25
размер турнира	4
максимальный размер выборки	8192
генное выражение длины головы	8
операции	+, -, *, /, exp, ln, cos, abs, mean, max
базовая фитнес-функция	Линейная SVM
фитнес-функции-примеси	[Логистическая регрессия, расстояние]

GP-DR и GP-MaL-MO были адаптированы для выполнения в нашем бенчмарке (это означает, что они должны реализовать интерфейс трансформера из библиотеки sklearn). После адаптации они были запущены в бенчмарке с конфигурациями по умолчанию. Единственное отличие в гиперпараметрах – это количество выходных признаков, которое было изменено на 2 там, где это было возможно.

Результатом GP-MaL-MO является фронт Парето с компромиссом между качеством решения и количеством измерений. В этом эксперименте использовано лучшее решение с двумя измерениями, если оно существовало, и объединение двух решений с одним измерением в противном случае.

В этом эксперименте мы сравниваем метрики линейной модели классификации (логистической регрессии), обученной и протестированной на признаках, сгенерированных различными алгоритмами уменьшения размерности. Все алгоритмы строят двумерное пространство признаков, поскольку это наиболее распространенный случай в реальной жизни, когда уменьшение размерности используется для того, чтобы сделать пространство признаков пригодным для визуализации (трехмерное пространство также может быть визуализировано, но построить понятную изометрическую визуализацию – более сложная задача). Линейная модель используется потому, что качество ее предсказаний показывает, насколько хорошо пространство признаков может быть линейно разделено. Линейно разделяемое пространство удобно для визуального анализа. Кроме того, линейные зависимости в целом более интерпретируемы и понятны человеку.

Генетические алгоритмы преобразуют пространство признаков с помощью преобразования, имеющего наибольшее значение фитнес-функции.

GP-DR не очень хорошо работает с достаточно большими наборами данных. Именно поэтому некоторые результаты для этого алгоритма отсутствуют. GP-MaL-MO также не может завершить вычисления на больших наборах данных.

Таблица 4

Сравнение прозрачных алгоритмов понижения размерности

Table 4

Comparison of the Transparent Dimension Reduction Algorithms

Набор данных	Baseline	GP-MaL-MO	GP-DR sammon euclid	GP-DR rank euclid	GP-DR sammon isomap	GP-DR rank isomap	GP-DR AE teacher	GP-DR AE fit	ГУРУ	Победитель
bank marketing	**0,589**	–	0,514	0,507	0,515	0,503	0,522	0,541	0,548	Baseline
bioresponse	0,514	0,571	0,500	0,500	0,507	0,508	0,498	–	**0,727**	ГУРУ
blood trans	0,500	0,509	0,523	0,513	0,517	0,517	0,518	0,523	**0,568**	ГУРУ
breast cancer wisconsin	0,867	0,851	0,785	0,702	0,774	0,649	0,886	0,870	**0,918**	ГУРУ
christine	0,583	0,574	0,527	0,530	0,529	0,534	0,577	–	**0,667**	ГУРУ
credit-g	**0,595**	0,562	0,571	0,560	0,571	0,560	0,581	0,581	**0,528**	Baseline
diabetes	0,707	0,604	0,552	0,546	0,624	0,548	0,610	0,579	**0,714**	ГУРУ
guillermo	0,507	–	–	–	–	–	–	–	**0,525**	ГУРУ
hyperplane	0,642	–	–	–	–	–	–	–	**0,672**	ГУРУ
ionosphere	0,653	0,578	0,607	0,583	0,579	0,552	0,544	0,559	**0,739**	ГУРУ
madelon	0,556	0,564	0,497	0,500	0,508	0,501	0,553	–	**0,602**	ГУРУ
sonar	**0,676**	0,583	–	–	–	–	0,589	0,602	**0,605**	Baseline

В качестве базового решения в этом эксперименте использовался отбор признаков с помощью линейной модели (величина коэффициентов логистической регрессии), поскольку выбор признаков можно рассматривать как примитивное прозрачное уменьшение размерности.

Анализ результатов

Как видно из табл. 4, ГУРУ превосходит другие генетические алгоритмы в этом эксперименте почти на всех наборах данных. Однако стоит отметить результаты на наборе данных credit-g, где ГУРУ показывает худший результат среди всех алгоритмов. Причиной этого, вероятно, является нетривиальная комбинация числовых и категориальных признаков, содержащих важную для классификации информацию. ГУРУ генерирует признаки без каких-либо знаний о категориальных признаках. Это приводит к эффектам, подобным этому, на тех наборах данных, где важна комбинация числовых и категориальных признаков.

Вероятно, такая же ситуация и с набором данных bank-marketing, поскольку в случае трехмерного пространства качество значительно лучше. В случае с набором данных sonar трудно делать предположения, потому что этот набор данных небольшой и имеет всего 208 объектов, и несколько других алгоритмов на таком малом датасете вообще не смогли успешно отработать.

Также стоит отметить, что ГУРУ – единственный из генетических алгоритмов смог успешно отработать на больших по количеству признаков датасетах guillermo и hyperplane.

Заключение

В этой работе мы попытались создать прозрачное решение для уменьшения размерности и представили ГУРУ. Алгоритм обеспечивает построение линейно сепарабельного низкоразмерного пространства признаков посредством конструирования признаков. Он работает с числовыми признаками, а сгенерированные признаки представлены в виде выражений генов, являющихся деревьями выражений, которые могут быть интерпретированы человеком. ГУРУ использует динамическую многокритериальную фитнес-функцию для оценки особей, которая позволяет строить разнообразные решения. Чтобы справиться с компромиссом между исследованием и эксплуатацией, конвейер алгоритмов разделен на две фазы: короткая фаза исследования с популяциями большого размера в начале и длинная фаза эксплуатации со значительно меньшими популяциями в дальнейшем. Эксперименты показали, что ГУРУ обеспечивает лучшее качество классификации по линейной модели среди аналогов в случае уменьшения размерности до двух измерений. Стоит отметить, что ГУРУ может быть использован как алгоритм инженерии признаков, который строит признаки, обеспечивающие высокое качество классификации и поддающиеся интерпретации. Он может быть использован аналитиком для генерации большого набора различных признаков и поиска среди них значимых, которые могут привести к пониманию данных. Исследование такого применения является одним из направлений будущей работы.

Список литературы

1. **M. H. ur Rehman, C. S. Liew, A. Abbas, P. P. Jayaraman, T. Y. Wah, S. U. Khan.** Big Data Reduction Methods: A Survey. Data Sci. Eng., 2016, vol. 1, no. 4, p. 265–284, DOI: 10.1007/s41019-016-0022-0.
2. **C. H. Yoon, R. Torrance, N. Scheinerman.** Machine learning in medicine: should the pursuit of enhanced interpretability be abandoned? J. Med. Ethics, 2022, vol. 48, no. 9, p. 581–585, DOI: 10.1136/medethics-2020-107102.

3. **P. Linardatos, V. Papastefanopoulos, S. Kotsiantis.** Explainable AI: A Review of Machine Learning Interpretability Methods. Entropy, 2021, vol. 23, no. 1, Art. no. 1, DOI: 10.3390/e23010018.
4. **Izenman.** Introduction to manifold learning. Wiley Interdiscip. Rev. Comput. Stat., 2012, vol. 4, DOI: 10.1002/wics.1222.
5. **H. Han, W. Li, J. Wang, G. Qin, X. Qin**. Enhance Explainability of Manifold Learning. Neurocomputing, 2022, vol. 500, DOI: 10.1016/j.neucom.2022.05.119.
6. **D. Elizondo, R. Birkenhead, M. Gámez, N. Rubio, E. Alfaro-Cortés.** Linear separability and classification complexity. Expert Syst. Appl., 2012, vol. 39, p. 7796–7807, DOI: 10.1016/j.eswa.2012.01.090.
7. **J. Koza, R. Poli.** Genetic Programming. in Search Methodologies, 2005, p. 127–164. DOI: 10.1007/0-387-28356-0_5.
8. **U.-M. O'Reilly, E. Hemberg.** Genetic programming: a tutorial introduction. 2021, p. 453. DOI: 10.1145/3449726.3461394.
9. **Vasuki.** Genetic Programming. 2020, p. 61–76. DOI: 10.1201/9780429289071-5.
10. **L. Kallel, B. Naudts, C. Reeves.** Properties of Fitness Functions and Search Landscapes. 2000, DOI: 10.1007/978-3-662-04448-3_8.
11. **M. Schmidt, H. Lipson.** Distilling Free-Form Natural Laws from Experimental Data. Science, 2009, vol. 324, no. 5923, p. 81–85, DOI: 10.1126/science.1165893.
12. **W. La Cava et al.** Contemporary Symbolic Regression Methods and their Relative Performance. in Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 2021, vol. 1.
13. L**. Sotto, P. Kaufmann, T. Atkinson, R. Kalkreuth, M. Basgalupp.** Graph representations in genetic programming. Genet. Program. Evolvable Mach., 2021, vol. 22, DOI: 10.1007/s10710-021-09413-9.
14. **P. Krtolica, P. Stanimirovic.** Reverse Polish notation method. Int. J. Comput. Math., 2004, vol. IJCM, p. 273–284, DOI: 10.1080/00207160410001660826.
15. **C. Ferreira.** Gene Expression Programming: a New Adaptive Algorithm for Solving Problems. arXiv, 2001. DOI: 10.48550/arXiv.cs/0102027.
16. **C. Ferreira.** Gene Expression Programming in Problem Solving. in Soft Computing and Industry: Recent Applications, Eds. London: Springer, 2002, p. 635–653. DOI: 10.1007/978-1-4471-0123-9_54.
17. **Jolliffe.** Principal Component Analysis. Springer: Berlin, Germany, 1986, vol. 87, p. 41–64, DOI: 10.1007/b98835.
18. **L. van der Maaten, G. Hinton.** Visualizing data using t-SNE. Journal of Machine Learning Research, 2008, vol. 9, p. 2579–2605.
19. **B. Hosseini, B. Hammer.** Interpretable Discriminative Dimensionality Reduction and Feature Selection on the Manifold. arXiv, arXiv:1909.09218, 2019 DOI: 10.48550/arXiv.1909.09218.
20. **T. Uriot, M. Virgolin, T. Alderliesten, P. Bosman.** On genetic programming representations and fitness functions for interpretable dimensionality reduction. arXiv, arXiv:2203.00528, 2022. DOI: 10.48550/arXiv.2203.00528.
21. **Lensen, M. Zhang, B. Xue.** Multi-Objective Genetic Programming for Manifold Learning: Balancing Quality and Dimensionality. Genet. Program. Evolvable Mach., 2020, vol. 21, no. 3, p. 399–431, DOI: 10.1007/s10710-020-09375-4.
22. **M. Virgolin, T. Alderliesten, P. A. N. Bosman.** On Explaining Machine Learning Models by Evolving Crucial and Compact Features. Swarm Evol. Comput., 2020 vol. 53, p. 100640, DOI: 10.1016/j.swevo.2019.100640.
23. **Lensen, B. Xue, M. Zhang.** Can Genetic Programming Do Manifold Learning Too? in Genetic Programming, Cham, 2019, p. 114–130. doi: 10.1007/978-3-030-16670-0_8.

24. **Q. Zhang, H. Li.** MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. IEEE Trans. Evol. Comput., 2007, vol. 11, no. 6, p. 712–731, DOI: 10.1109/TEVC.2007.892759.
25. **S. Roweis, L. Saul.** Nonlinear Dimensionality Reduction by Locally Linear Embedding. Science, 2001, vol. 290, p. 2323–6, DOI: 10.1126/science.290.5500.2323.
26. **B. K. Tripathy, S. Anveshrithaa, S. Ghela.** Multidimensional Scaling (MDS). 2021, p. 41–51. DOI: 10.1201/9781003190554-6.
27. **L. McInnes, J. Healy, J. Melville.** UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv, Sep. 17, 2020. DOI: 10.48550/arXiv.1802.03426.
28. **C. Cortes, V. Vapnik.** Support vector machines. Mach. Learn., 1995, vol. 20, p. 273–293.
29. **Vanschoren, J. N. van Rijn, B. Bischl, L. Torgo.** OpenML: networked science in machine learning. ACM SIGKDD Explor. Newsl., 2014, vol. 15, no. 2, p. 49–60, DOI: 10.1145/2641190.2641198.
30. **S. Moro, P. Cortez, R. Laureano.** Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology. 2011.
31. **Yeh, K.-J. Yang, T.-M. Ting.** Knowledge discovery on RFM model using Bernoulli sequence. Expert Syst Appl, 2009, vol. 36, p. 5866–5871, DOI: 10.1016/j.eswa.2008.07.018.
32. **Bennett, O. L. Mangasarian.** Robust Linear Programming Discrimination Of Two Linearly Inseparable Sets. Optim. Methods Softw., 2002, vol. 1, DOI: 10.1080/10556789208805504.
33. **D. Dua, C. Graff.** UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences, 2019. [Online]. Available: http://archive.ics.uci.edu/ml
34. **V. Sigillito, S. Wing, L. Hutton, K. Baker.** Classification of radar returns from the ionosphere using neural networks. Johns Hopkins APL Tech. Dig. Appl. Phys. Lab., 1989, vol. 10.
35. **Guyon, S. Gunn, A. Ben-Hur, G. Dror.** Result Analysis of the NIPS 2003 Feature Selection Challenge, vol. 17. 2004.
36. **R. P. Gorman, T. Sejnowski.** Analysis of hidden units in a layered network trained to classify sonar targets. Neural Netw., 1988, vol. 1, p. 75–89, DOI: 10.1016/0893-6080(88)90023-8

References

1. **ur Rehman M. H., Liew C. S., Abbas A., Jayaraman P. P., T Wah. Y., Khan S. U.** Big Data Reduction Methods: A Survey // Data Sci. Eng. 2016. Vol. 1, no. 4. Pp. 265–284. DOI 10.1007/s41019-016-0022-0
2. **Yoon C. H., Torrance R., Scheinerman N..** Machine learning in medicine: should the pursuit of enhanced interpretability be abandoned? // J. Med. Ethics. 2022. Vol. 48, no. 9. Pp. 581–585. DOI 10.1136/medethics-2020-107102
3. **Linardatos P., Papastefanopoulos V., Kotsiantis S.** Explainable AI: A Review of Machine Learning Interpretability Methods // Entropy. 2021. Vol. 23, no. 1. Art. 1. DOI 10.3390/e23010018
4. **Izenman.** Introduction to manifold learning // Wiley Interdiscip. Rev. Comput. Stat. 2012. Vol. 4. DOI 10.1002/wics.1222
5. **Han H., Li W., Wang J., Qin G., Qin X..** Enhance Explainability of Manifold Learning // Neurocomputing. 2022. Vol. 500. DOI 10.1016/j.neucom.2022.05.119
6. **Elizondo D., Birkenhead R., Gámez M., Rubio N., Alfaro-Cortés E.** Linear separability and classification complexity // Expert Syst. Appl. 2012. Vol. 39. Pp. 7796–7807. DOI 10.1016/j.eswa.2012.01.090
7. **Koza J., Poli R..** Genetic Programming / Search Methodologies, 2005. Pp. 127–164. DOI 10.1007/0-387-28356-0_5
8. **O'Reilly U.-M., Hemberg E.** Genetic programming: a tutorial introduction. 2021, p. 453. DOI 10.1145/3449726.3461394
9. **Vasuki.** Genetic Programming. 2020. Pp. 61–76. DOI 10.1201/9780429289071-5
10. **Kallel L., Naudts B., Reeves C.** Properties of Fitness Functions and Search Landscapes. 2000. DOI 10.1007/978-3-662-04448-3_8

11. **Schmidt M., Lipson H.** Distilling Free-Form Natural Laws from Experimental Data // Science. 2009. Vol. 324, no. 5923. Pp. 81–85. DOI 10.1126/science.1165893
12. **La Cava W. et al.** Contemporary Symbolic Regression Methods and their Relative Performance / Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks, 2021, vol. 1.
13. **Sotto L., Kaufmann P., Atkinson T., Kalkreuth R., Basgalupp M.** Graph representations in genetic programming // Genet. Program. Evolvable Mach., 2021, vol. 22. DOI 10.1007/s10710-021-09413-9
14. Kr**tolica P., Stanimirovic P..** Reverse Polish notation method // Int. J. Comput. Math., 2004, vol. IJCM, p. 273–284. DOI 10.1080/00207160410001660826
15. **Ferreira C.** Gene Expression Programming: a New Adaptive Algorithm for Solving Problems. arXiv, 2001. DOI 10.48550/arXiv.cs/0102027
16. **Ferreira C.** Gene Expression Programming in Problem Solving. in Soft Computing and Industry: Recent Applications. London: Springer, 2002, pp. 635–653. DOI 10.1007/978-1-4471-0123-9_54
17. **Jolliffe.** Principal Component Analysis. Springer: Berlin, Germany, 1986. Vol. 87, pp. 41–64. DOI 10.1007/b98835
18. **van der Maaten L., Hinton G.** Visualizing data using t-SNE // Journal of Machine Learning Research, 2008. Vol. 9, pp. 2579–2605.
19. **Hosseini B., Hammer B.** Interpretable Discriminative Dimensionality Reduction and Feature Selection on the Manifold. arXiv, arXiv:1909.09218, 2019 doi: 10.48550/arXiv.1909.09218.
20. **Uriot T., Virgolin M., Alderliesten T., Bosman P.** On genetic programming representations and fitness functions for interpretable dimensionality reduction. arXiv, arXiv:2203.00528, 2022. DOI 10.48550/arXiv.2203.00528
21. **Lensen, M., Zhang, B. Xue.** Multi-Objective Genetic Programming for Manifold Learning: Balancing Quality and Dimensionality. Genet. Program. Evolvable Mach. 2020. Vol. 21, no. 3, pp. 399–431. DOI 10.1007/s10710-020-09375-4
22. **Virgolin M., Alderliesten T., Bosman P. A. N.** On Explaining Machine Learning Models by Evolving Crucial and Compact Features. Swarm Evol. Comput. 2020. vol. 53, p. 100640. DOI 10.1016/j.swevo.2019.100640
23. **Lensen, B. Xue, M. Zhang.** Can Genetic Programming Do Manifold Learning Too? / Genetic Programming, Cham, 2019, p. 114–130. DOI 10.1007/978-3-030-16670-0_8
24. **Zhang Q., Li H.** MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition // IEEE Trans. Evol. Comput., 2007, vol. 11, no. 6, p. 712–731. DOI 10.1109/TEVC.2007.892759
25. **Roweis S., Saul L..** Nonlinear Dimensionality Reduction by Locally Linear Embedding // Science, 2001, vol. 290, p. 2323–6. DOI 10.1126/science.290.5500.2323
26. **Tripathy B. K., Anveshrithaa S., Ghela S.** Multidimensional Scaling (MDS). 2021, p. 41–51. DOI 10.1201/9781003190554-6
27. **McInnes L., Healy J., Melville J.** UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv, Sep. 17, 2020. DOI 10.48550/arXiv.1802.03426
28. **Cortes C., Vapnik V.** Support vector machines // Mach. Learn., 1995, vol. 20, pp. 273–293.
29. **Vanschoren J., van Rijn N., Bischl B., Torgo L.** OpenML: networked science in machine learning. ACM SIGKDD Explor. Newsl., 2014, vol. 15, no. 2, p. 49–60. DOI 10.1145/2641190.2641198
30. **Moro S., Cortez P., Laureano R.** Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology. 2011.
31. **Yeh, Yang K.-J., Ting T.-M.** Knowledge discovery on RFM model using Bernoulli sequence. Expert Syst Appl, 2009, vol. 36, p. 5866–5871. DOI 10.1016/j.eswa.2008.07.018
32. **Bennett, Mangasarian O. L..** Robust Linear Programming Discrimination of Two Linearly Inseparable Sets. Optim. Methods Softw., 2002, vol. 1. DOI 10.1080/10556789208805504

33. **Dua D., Graff C.** UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences, 2019. [Online]. URL: http://archive.ics.uci.edu/ml.
34. **Sigillito V., Wing S., Hutton L., Baker K.** Classification of radar returns from the ionosphere using neural networks. Johns Hopkins APL Tech. Dig. Appl. Phys. Lab., 1989, vol. 10.
35. **Guyon, Gunn S., Ben-Hur A., Dror G.** Result Analysis of the NIPS 2003 Feature Selection Challenge, 2004. Vol. 17.
36. **Gorman R. P., Sejnowski T.** Analysis of hidden units in a layered network trained to classify sonar targets. Neural Netw., 1988, vol. 1, p. 75–89. DOI 10.1016/0893-6080(88)90023-8

Информация об авторе

Радеев Никита Андреевич, магистрант кафедры общей информатики факультета информационных технологий НГУ

Information about the Author

Nikita A. Radeev, master of General Informatics Department, faculty of Information Technologies, Novosibirsk state University (Novosibirsk, Russian Federation)

Статья поступила в редакцию 29.03.2023;
одобрена после рецензирования 25.04.2023; принята к публикации 25.04.2023
The article was submitted 29.03.2023;
approved after reviewing 25.04.2023; accepted for publication 25.04.2023

Научная статья

УДК 004.89
DOI 10.25205/1818-7900-2023-21-1-62-72

Разработка системы выявления аномалий на основе распределенной трассировки логов

Даниил Александрович Худяков

Новосибирский государственный университет
Новосибирск, Россия

khudaikoff@gmail.com, https://orcid.org/0009-0005-2747-8535

Аннотация

Разработчики программных систем должны оперативно реагировать на сбои, чтобы избежать репутационных и финансовых потерь для своих заказчиков. Поэтому важно своевременно обнаруживать поведенческие аномалии в работе программных систем. На данный момент активно развиваются различные средства для автоматического мониторинга работы систем, однако главным инструментом для анализа сбоев являются логи. Логи содержат информацию о работе системы в различных точках исполнения. Современные системы часто имеют распределенную микросервисную архитектуру, что значительно усложняет задачу анализа логов. Логи таких систем собираются централизованно из разных микросервисов, образуя огромный поток информации, которую очень сложно анализировать вручную. Однако проблему идентификации логов, относящихся к конкретному запросу в систему, решает распределенная трассировка, использование которой открывает широкие возможности для внедрения автоматического анализа. Уже существует множество решений для обнаружения аномалий в логах, однако они не используют преимущества распределенной трассировки. Статья посвящена решению задачи обнаружения поведенческих аномалий в работе распределенных программных систем на основе автоматического анализа трассировок логов. Решение основано на синтезе методов машинного обучения. Цепочки логов проходят предобработку, а также очистку с использованием методов процессной аналитики. Далее производится векторизация и кластеризация сообщений логов. После чего для анализа отклонений в последовательностях обработанных логов применяется сеть долгой краткосрочной памяти (LSTM). В результате проведенной работы был разработан и протестирован прототип системы обнаружения аномалий.

Ключевые слова

микросервисная архитектура, анализ логов, обнаружение аномалий, распределенная трассировка, машинное обучение, процессная аналитика

Для цитирования

Худяков Д. А. Разработка системы выявления аномалий на основе распределенной трассировки логов // Вестник НГУ. Серия: Информационные технологии. 2023. Т. 21, № 1. С. 62–72. DOI 10.25205/1818-7900-2023-21-1-62-72

Development of Anomaly Detection System Based on Distributed Log Tracing

Daniil A. Khudyakov

Novosibirsk State University
Novosibirsk, Russian Federation

khudaikoff@gmail.com, https://orcid.org/0009-0005-2747-8535

Abstract

Software system developers must respond quickly to failures in order to avoid reputational and financial losses for their customers. Therefore, it is important to detect behavioral anomalies in the operation of software systems in a timely manner. At the moment, various tools for automatic monitoring of systems are being actively developed, but logs are the main tool for analyzing failures. Logs contain information about the operation of the system at various points of execution. Modern systems often have a distributed microservice architecture, which significantly complicates the task of analyzing logs. Logs of such systems are collected centrally from different microservices, forming a huge flow of information that is very difficult to analyze manually. However, the problem of identifying logs related to a specific request to the system is solved by distributed tracing, the use of which opens up wide opportunities for the introduction of automatic analysis. There are already many solutions for detecting anomalies in logs, but they do not take advantage of distributed tracing. The article is considered to solving the problem of detecting behavioral anomalies in the work of distributed software systems based on automatic analysis of log traces. The solution is based on the synthesis of machine learning methods. Log traces are preprocessed and cleaned using process mining methods. Next, vectorization and clustering of log messages is performed. After that, a long short-term memory network (LSTM) is used to analyze deviations in the sequences of processed logs. As a result of the work performed, a prototype of the anomaly detection system was developed and tested.

Keywords

micro service architecture, log analysis, anomaly detection, distributed tracing, machine learning, process mining

For citation

Khudyakov D. A. Development of Anomaly Detection System Based on Distributed Log Tracing. *Vestnik NSU. Series: Information Technologies*, 2023, vol. 21, no. 1, pp. 62–72. (in Russ.) DOI 10.25205/1818-7900-2023-21-1-62-72

Введение

Большинство существующих программных систем подвержены различным сбоям из-за сложности и неизбежных недостатков программного и аппаратного обеспечения системы. Такие сбои приводят к снижению надежности, высоким финансовым затратам и могут повлиять на критически важные приложения. Таким образом, потеря контроля не допускается ни для одной системы или инфраструктуры, поскольку качество обслуживания имеет большое значение. Автоматизация обнаружения аномалий значительно помогает командам разработчиков сократить время, затрачиваемое на поиск сбоев. Это подразумевает обнаружение и распознавание паттернов, которые не соответствуют ожидаемому поведению системы. Необходимым условием для устранения аномалий [1], возникающих в любой системе, является наличие наблюдаемых данных [2] о ее работе. Наиболее полезными системными данными являются логи. Основное назначение логов – записывать состояние системы и важные события в различных критических точках выполнения. Логи являются основой многих решений, которые помогают решить проблему автоматизации обнаружения аномалий с помощью машинного обучения [3, 4].

В отличие от большинства проблем и задач машинного обучения, обнаружение аномалий касается непредсказуемых, неопределенных и редких событий, что приводит к определенным трудностям [5–7]. Современные программные системы часто имеют распределенную архитектуру, основанную на концепции микросервиса, которая обеспечивает быструю, частую и надежную доставку больших и сложных приложений. Эта парадигма в разработке программного обеспечения обеспечивает гибкость систем, но также значительно увеличивает их сложность

[8, 9]. Микросервисная архитектура предполагает разбиение системы на слабо связанные, независимо развертываемые и масштабируемые компоненты, отвечающие за узкий набор функций. Сами компоненты, как правило, представляют собой многопоточные приложения, способные обрабатывать множество запросов параллельно. Однако логи распределенных систем собираются централизованно из разных микросервисов в единое хранилище. В результате происходит их сильное перемешивание, из-за чего становится невозможным понять, например, какие логи относятся к конкретному запросу пользователя или внутренней задаче [10]. В результате анализ логов распределенных систем значительно усложняется.

Главной особенностью сбора логов распределенных систем является их автоматическое объединение в цепочки, благодаря использованию распределенной трассировки[1]. В этой статье под цепочкой (трассой) подразумевается группа логов, сгенерированных по одному пользовательскому запросу. Цепочки позволяют избавиться от неопределенности в логах, которая затрудняет их анализ. Однако существующие решения не используют преимущества распределенной трассировки, поэтому их применение в распределенных системах может иметь крайне ограниченный результат. Основной целью данного исследования является разработка нового решения, использующего возможности распределенной трассировки логов для решения проблемы автоматизации обнаружения аномалий.

В первом параграфе статьи описывается процесс сбора логов распределенных систем. Второй параграф посвящен описанию существующих решений для поиска аномалий на основе анализа логов. В третьем параграфе описывается общая архитектура предлагаемого решения. Четвертый параграф детализирует описание процесса предварительной обработки данных. В пятом параграфе описывается принцип работы сети LSTM и особенности ее настройки. В шестом параграфе приведены результаты работы над прототипом решения, протестированном на сгенерированных синтетических данных. В статье также обсуждаются дальнейшие направления исследований.

1. Распределенная трассировка

Для того чтобы с логами систем, имеющих микросервисную архитектуру, можно было работать, применяется распределенная трассировка – диагностическая методика, используемая для профилирования и мониторинга приложений, построенных с использованием архитектуры микросервисов. Распределенная трассировка открывает широкие возможности для анализа логов. В частности, разработчики и инженеры технической поддержки при ручном разборе инцидентов могут видеть контекст ошибок, что упрощает поиск причины их происхождения. Распределенная трассировка реализуется во многих современных популярных фреймворках. Например, инструмент Spring Cloud Sleuth[2] для Spring Framework. Схема его работы изображена на рис. 1.

Spring Cloud Sleuth осуществляет распределенную трассировку отдельных запросов к системе, прослеживая путь потока исполнения при обработке каждого запроса, т. е. при его прохождении через микросервисы. В процессе обработки запроса логи помечаются, благодаря чему появляется возможность легко их сгруппировать. Такую группу логов можно назвать цепочкой или трассой. Каждой цепочке присваивается идентификатор – **traceId**. Внутри цепочки логи также группируются, если внутри системы происходят запросы между микросервисами. После каждого такого запроса генерируется новый идентификатор **spanId**, который присваивается всем логам до следующего внутреннего запроса.

[1] A Quick Introduction to Distributed Tracing. URL: https://newrelic.com/sites/default/files/2021-08/quick-introduction-to-distributed-tracing.pdf

[2] Spring Cloud Sleuth. URL: https://cloud.spring.io/spring-cloud-sleuth/reference/html/

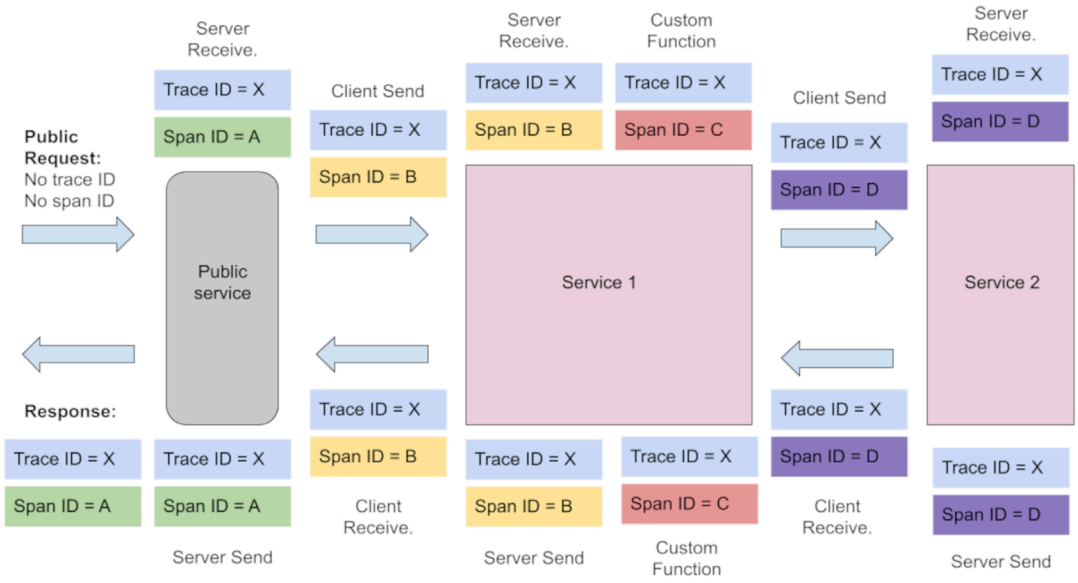


Рис. 1. Схема работы Spring Cloud Sleuth
Fig. 1. How Spring Cloud Sleuth operates

Далее рассмотрим процесс сбора и хранения логов. На данный момент в промышленной разработке повсеместно используется набор технологий под аббревиатурой ELK[3]. В данный набор входят такие продукты, как Elasticsearch, Logstash и Kibana. ELK предоставляет возможность собирать логи различных систем, хранить и визуализировать их. Elasticsearch – это распределенный поисковый и аналитический движок на базе Apache Lucene, а также хранилище данных. Logstash – это инструмент, предназначенный для сбора, фильтрации и последующего перенаправления логов систем в их конечное хранилище. Kibana – это инструмент визуализации логов, который полезен разработчикам и инженерам технической поддержки для разбора инцидентов в работе систем. Logstash позволяет гибко кастомизировать логирование, однако большинство разработчиков старается придерживаться хотя бы минимально необходимого набора атрибутов для **каждого лога**:

1) timestamp – время возникновения лога
2) level – DEBUG, INFO, и т.д.
3) applicationName – название микросервиса
4) username – инициатор запроса
5) traceId – id цепочки
6) spanId – id спана
7) message – сообщение лога
8) METHOD – название запроса в системе

Именно на основе данного набора атрибутов и будет основываться работа представленного в статье решения.

2. Существующие решения

Далее перейдем к краткому обзору существующих подходов к детектированию аномалий в логах и оценим их применимость к логам распределенных систем. Исследования в области детектирования аномалий можно разделить на два крупных направления.

[3] What is the ELK stack? URL: https://aws.amazon.com/what-is/elk-stack/

Использование статистических методов машинного обучения для анализа так называемых, message count векторов (MCV) [11, 12]. MCV аналогичен понятию мешка слов из обработки естественного языка. MVC сопоставляется цепочке логов, прошедшей стадию препроцессинга, и отображает количество произошедших в ней сообщений того или иного типа. Для анализа MVC используются различные подходы. Один самых популярных – это использование метода главных компонент (PCA). У данного подхода имеется ряд минусов, которые мешают его использованию в распределенных системах. Рассмотрим их:

1) Логи нестабильны. Системы постоянно обновляются, появляются новые логи, а старые выходят из использования. При этом размерность MCV соответственно должна изменяться. Модель, соответственно, должна переобучаться с новыми параметрами из-за чего использование статических методов не подходит для задачи онлайн-детектирования.

2) Не учитывается контекстная информация. При анализе используется только частота появлений тех или иных сообщений в последовательности логов. Контекстная информация при этом никак не учитывается, из-за чего не удается определить степень важности отдельного сообщения в рамках контекста.

Другой подход основан на использовании методов глубокого машинного обучения. Главное направление в этой области – это анализ цепочек логов с помощью глубоких нейронных сетей долгой краткосрочной памяти (LSTM) [13, 14]. Предсказание при этом подходе является вероятностным и базируется на анализе цепочки логов заданной длины, предшествующей рассматриваемому логу, называемой окном. Лог считается аномальным, если он не попадает в заданное число наиболее вероятных, предсказанных моделью.

Однако прямое использование одного из существующих решений на основе LSTM-сетей применительно к логам распределенных может не дать хороших результатов в силу следующих причин:

1) Использование готовых парсеров логов может не сработать. Необходимо учитывать специфику происхождения логов. Часто более производительным и точным вариантом препроцессинга будет использование собственного решения, адаптированного, например, для сообщений в формате JSON или XML.

2) Не решается проблема перемешивания логов. Если брать логи напрямую в порядке их записи в хранилище, то сформированные окна логов, на основе которых производится анализ, потенциально будут содержать сообщения из разных запросов, от разных пользователей, что лишает их смысла.

3) Проблема выбора ширины окна. Крайне важным гиперпараметром моделей на основе LSTM-сетей является ширина окна рассматриваемых логов. Однако цепочки логов имеют самую разную длину. И решение, использующее одну заданную ширину окна, смогло бы обнаружить лишь некоторые из закономерностей в логах, что сильно снижает его точность.

Перечисленные проблемы создают предпосылки для создания нового решения для детектирования аномалий в логах, которое могло бы учитывать специфику сбора логов распределенных систем.

3. Архитектура решения

Далее рассмотрим общую архитектуру предлагаемого решения. На рис. 2 схематически изображена схема его работы. На вход в систему поступает множество «сырых» логов, каждый из которых имеет описанную структуру. Далее на основе **traceId** производится формирование цепочек логов, после чего цепочки группируются в зависимости от «активности», т. е. в зависимости от типа запроса в систему (определяется на основе поля **method** первого лога в цепочке). Далее для каждой активности последовательно производится построение модели процесса, очистка от шума, векторизация и последующая кластеризация сообщений отдельных логов в цепочках. После чего преобразованные цепочки поступают на вход LSTM-модели для ее обучения.

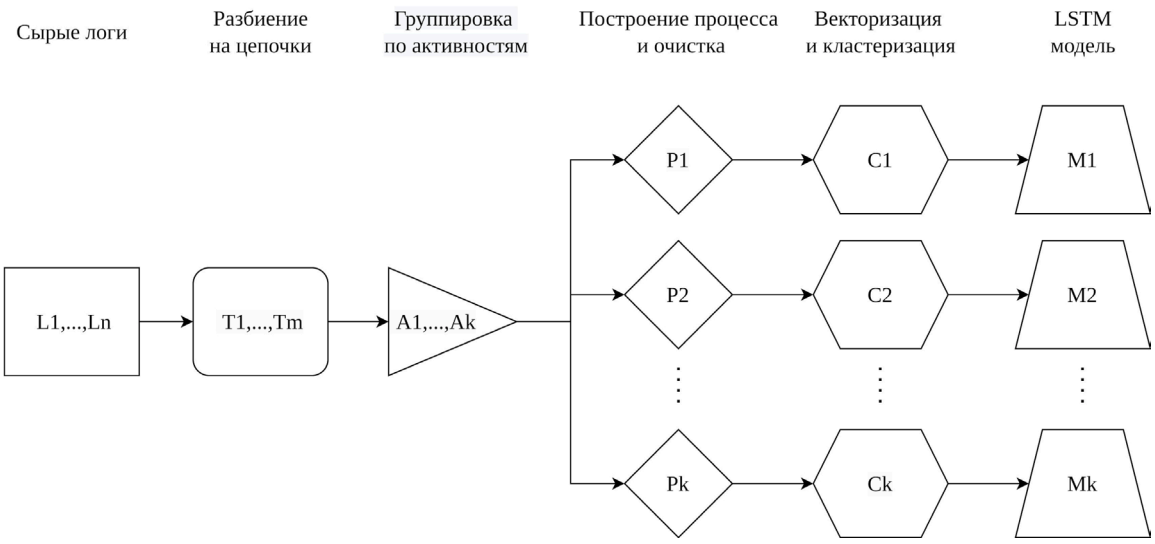


Рис. 2. Процесс обучения системы детектирования аномалий
Fig. 2. Process of training the anomaly detection system

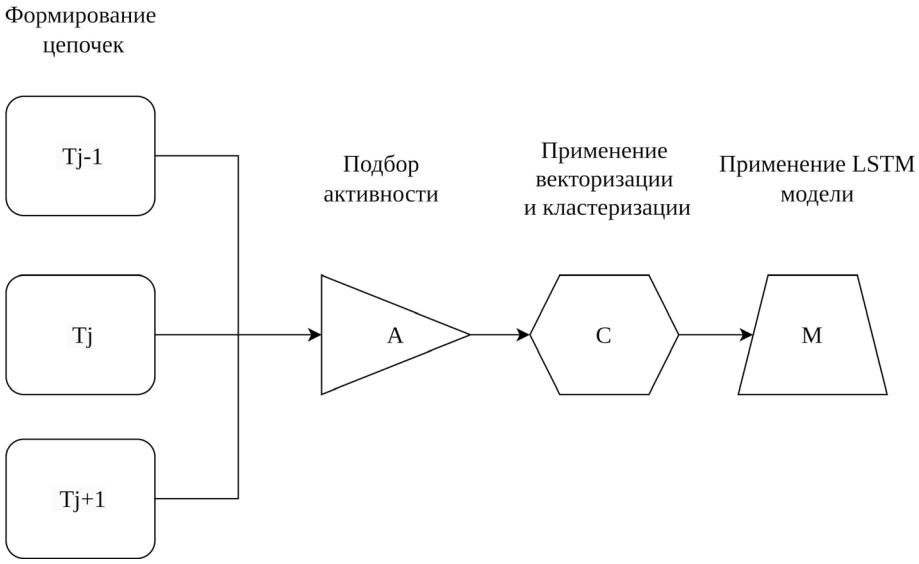


Рис. 3. Процесс предсказания аномалий
Fig. 3. Anomaly prediction process

Процесс предсказания аномалий для новых данных изображен на рис. 3. Из множества новых рассматриваемых логов выделяются цепочки, после чего каждая цепочка анализируется отдельно. Для цепочки подбирается активность, она обрабатывается с использованием ранее обученных моделей векторизации и кластеризации. И далее преобразованная цепочка поступает на вход LSTM-модели, которая делает предсказание для каждого отдельного взятого лога.

4. Предобработка данных

После группировки цепочек для каждой активности производится построение модели процесса. Как описывалось выше, принимающий запрос от пользователя микросервис внутри системы может вызывать запросы у других микросервисов, которые, в свою очередь, также

могут вызывать запросы по цепочке. Таким образом, формируется цепочка вызовов запросов в системе. Важно, что элементы множества цепочек, относящихся к одной активности (представляющих один тип запроса в систему), могут сильно различаться с точки зрения последовательности внутренних запросов т. к. последовательность может зависеть от параметров запроса. Построив все варианты последовательностей внутренних запросов для активности, мы можем получить описание процесса данной активности. В частности, такой процесс можно изобразить, построив взвешенный ориентированный граф (directed flow graph), где вершины – это названия запросов в системе, а ребра обозначают последовательный вызов пары запросов, вес каждого ребра – количество последовательных вызовов. Такой граф интересен с точки зрения применения к нему методов процессной аналитики (process mining).

При дальнейшем обучении LSTM-модели важно, чтобы в тренировочной выборке присутствовали только сильные закономерности, для этого нам нужно попытаться избавиться от шума. Для этого было решено использовать алгоритм Inductive Miner Infrequent [15] из библиотеки pm4py[4]. Применение этого алгоритма к взвешенному ориентированному графу в результате дает дерево переходов, на соответствие которых, мы можем проверить каждую цепочку, относящуюся к активности. Данный алгоритм позволяет конфигурировать параметр границы шума (noise_threshold), его настройка приводит к изменению дерева переходов т. к. из него выпадают редко происходящие переходы. Таким образом, выбрав значение параметра границы шума и построив дерево переходов, мы можем отбросить часть цепочек логов, относящихся к активности, которые не могут быть выровнены по построенному дереву переходов. В итоге данный метод был использован для очистки от шума тренировочной выборки.

Далее после формирования тренировочной выборки для каждой активности производится очистка сообщений всех логов от параметров. Для этого из сообщений логов исключатся все цифры, специальные символы, а также ряд известных заранее паттернов. В частности, известно, что современные веб-сервисы для передачи данных чаще всего используют сообщения в форматах JSON и XML. Можно сказать, что данные целиком являются параметрами, поэтому также должны быть исключены из сообщения лога, что и было сделано в рамках реализации решения.

После очистки сообщений происходит их векторизация. Для этого применяется алгоритм Doc2Vec [16]. Выбор этого алгоритма обоснован тем, что в результате его применения получаются векторы заданной размерности, которую можно задать небольшой, что ускорит дальнейшую кластеризацию. Другим важным аргументов является то, что полученную в результате применения алгоритма модель можно интеративно дообучать без полного пересчета (в отличие от, например, TF-IDF), что крайне важно для реализации онлайн-детектирования аномалий, когда система постоянно подкрепляется новыми данными. После векторизации сообщения логов кластеризуются, чтобы в результате преобразовать последовательности сообщений логов в последовательности чисел (каждому логу в результате кластеризации назначается числовая метка). Для кластеризации используется алгоритм KMeans. Число соседей подбирается путем рассмотрения длин цепочек активности, рассматриваются варианты в диапазоне от минимальной до максимальной длины цепочки. Лучший вариант выбирается на основе расчета коэффициента силуэта.

5. Детектирования аномалий

После завершения этапа препроцессинга цепочки, представленные числовыми последовательностями, используются для обучения нейронной сети LSTM. LSTM (long short-term memory) нейронная сеть – это рекуррентная нейронная сеть, способная к обучению долговременным зависимостям. Такие модели хорошо подходят для анализа событий, продолжающихся во времени. Кроме того, данная нейронная сеть обучается без учителя. Такой вариант

[4] pm4py documentation. URL: https://pm4py.fit.fraunhofer.de/documentation

обучения является единственно возможным вариантом при работе с логами распределенных систем, так как произвести их ручную разметку, когда логов становится огромное количество, невозможно.

Гиперпараметром модели является ширина окна **h**, т. е. количество событий (в нашем случае логов), предшествующих рассматриваемому событию, на основе которых делается предсказание. LSTM-модель обучается искать закономерности в последовательностях, после чего выдает вероятность для рассматриваемого события быть каждым из известных событий. После чего задается порог **k** (также гиперпараметр), и если рассматриваемое событие не попадает в список **k** наиболее вероятных предсказанных событий, то оно считается аномальным.

6. Результаты

В соответствии с описанным алгоритмом был реализован прототип системы детектирования аномалий. Также для его тестирования был разработан алгоритм генерации синтетических данных, результатом работы которого являются тренировочная и тестовая выборки, сгенерированные на основе линейной последовательности логов, представленной графом, и в которую специальным образом внесены отклонения, которые считаются аномалиями. Для построения графа использовалась библиотека NetworkX[5]. Пример последовательности, для которой производились расчеты, изображен на рис. 4. Выборки генерируются путем обхода графа. Тренировочная выборка генерируется по узлам, которые не считаются аномальными (на рисунке выделены зеленым), в тестовой же выборке часть примеров генерируется с проходом по узлам, представляющим аномалии (на рисунке выделены красным). Для генерации сообщений логов использовалась нейронная сеть gpt-neo[6].

Параметры генерации данных для тестирования отображены в табл. 1.

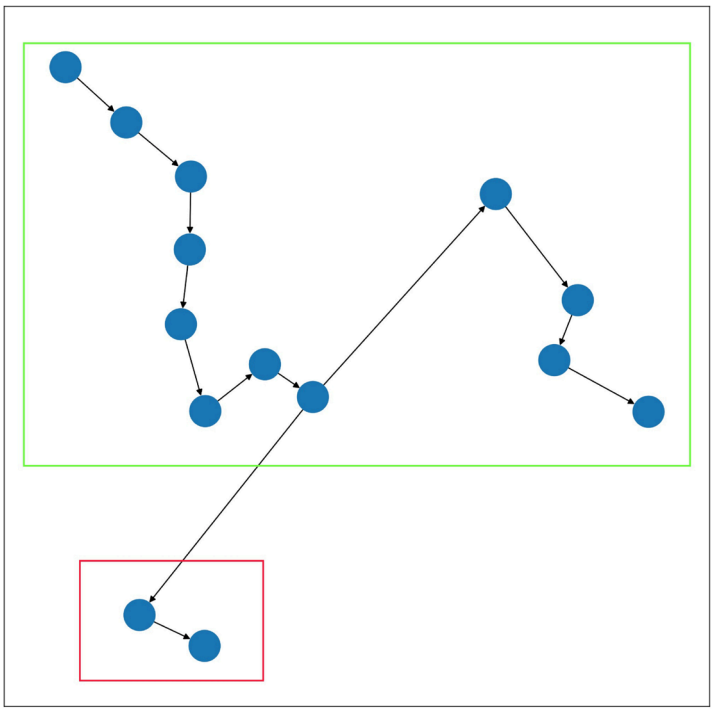


Рис. 4. Граф для генерации выборок
Fig. 4. Graph for generating samples

[5] NetworkX. URL: https://networkx.org/
[6] GPT Neo. URL: https://github.com/EleutherAI/gpt-neo

Таблица 1

Параметры генерации выборок

Table 1

Sample Generation Options

Размер тренировочной выборки	300
Размер тестовой выборки	100
Процент аномальных примеров в тестовой выборке	25%
Длина цепочки логов	12
Количество спанов	3
Максимальная длина сообщения лога	8

Таблица 2

Результаты тестирования

Table 2

Test Results

Точность	**Полнота**	**F1 – мера**
0.969	0.64	0.77

Результаты расчета метрик качества классификации (мы решаем задачу бинарной классификации) на тестовой выборке отображены в табл. 2. Как можно увидеть, прототип хорошо справляется с поиском аномалий. Высокая точность (precision) показывает, что система хорошо умеет определять именно аномальные логи, однако результаты могут быть улучшены с точки зрения полноты (recall).

Заключение

Микросервисная архитектура современных программных систем усложняет задачу обнаружения сбоев, главным инструментом для поиска которых являются логи. Распределенная трассировка логов открывает большие возможности для их анализа. Однако существующие решения для автоматического обнаружения аномалий в логах не используют преимущества распределенной трассировки. В данной работе предлагается новое решение для поиска аномалий в логах распределенных систем, имеющих микросервисную архитектуру. В ходе работы была спроектирована архитектура системы выявления аномалий на основе логов, собранных с использованием распределенной трассировки. В соответствии с архитектурой был реализован прототип. Для тестирования прототипа был реализован алгоритм генерации синтетических размеченных данных. Тестирование прототипа на сгенерированных тестовых данных показало, что система хорошо справляется с поиском аномалий в логах с точки точности прогнозирования, однако алгоритм может быть улучшен с точки зрения полноты классификации аномалий.

В дальнейшем планируется протестировать работу системы на более сложных синтетических данных, в которые будет вноситься больше шума, свойственного реальным системам. Также планируется работа над улучшением метрик качества классификации аномалий, в том числе путем настройки гиперпараметров моделей. Главным же направлением развития системы будет ее доработка для реализации возможности итеративного дообучения, что необходимо для решения задачи онлайн-детектирования аномалий. После доработок система будет способ-

на автоматически дообучаться на новых данных через каждый заданный промежуток времени, таким образом адаптируясь под изменения системы, являющейся источником логов.

Список литературы/References

1. **Chandola V., Banerjee A., Kumar V.** Anomaly Detection: A Survey. ACM Computing Surveys (CSUR) 41.3, 2009. p. 1–58. DOI: 10.1145/1541880.1541882
2. **Sridharan C. D**istributed Systems Observability: A Guide to Building Robust Systems. O'Reilly Media, 2018.
3. **Sultana N., Chilamkurti N., Peng, W., Alhadad R.** Survey on SDN Based Network Intrusion Detection System Using Machine Learning approaches. Peer-to-Peer Networking and Applications, 2018. p. 1–9. DOI: 10.1007/s12083-017-0630-0
4. **Pang G., Shen C., Cao L., van den Hengel A.** Deep Learning for Anomaly Detection: A Review, 2020. arXiv: 2007.02500. DOI: 10.1145/3439950
5. **Palchunov D. E., Yakhyaeva G. E.** Integration of Fuzzy Model Theory and FCA for Big Data Mining. SIBIRCON 2019 – International Multi-Conference on Engineering, Computer and Information Sciences, Proceedings, 2019. p. 961–966. DOI: 10.1109/sibircon48586.2019.8958216
6. **Yakhyaeva G. E.** Application of Boolean Valued and Fuzzy Model Theory for Knowledge Base Development. SIBIRCON 2019 - International Multi-Conference on Engineering, Computer and Information Sciences, Proceedings, 2019. p. 868–871. DOI: 10.1109/sibircon48586.2019.8958245
7. **Palchunov D. E.,. Tishkovsky D. E,. Tishkovskaya S. V, Yakhyaeva G. E.** Combining logical and statistical rule reasoning and verification for medical applications. Proceedings – 2017 International Multi-Conference on Engineering, Computer and Information Sciences, SIBIRCON 2017, 2017, p. 309-313. DOI: 10.1109/sibircon.2017.8109895
8. **Esposito C., Castiglione A., Choo K. R.** Challenges in Delivering Software in the Cloud as Microservices. IEEE Cloud Computing 3.5, 2016, p. 10–14. DOI: 10.1109/mcc.2016.105
9. **Gunawi H. S., Hao M., Leesatapornwongsa T., Patana-anake T., Do T., Adityatama J., Eliazar K. J., Laksono A., Lukman J. F., Martin V.** What Bugs Live in the Cloud? A Study of 3000+ Issues in Cloud Systems. Proceedings of the ACM Symposium on Cloud Computing, 2014. p. 1–14. DOI: 10.1145/2670979.2670986
10. **Beschastnikh I., Wang P., Brun Y., Ernst M. D.** Debugging distributed systems: Challenges and options for validation and debugging. Communications of the ACM, vol. 59, no. 8, Aug. 2016. p. 32-37. DOI: 10.1145/2927299.2940294
11. **Xu W., Huang L., Fox A., Patterson D., Jordan M. I.** Detecting large-scale system problems by mining console logs. Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles (SOSP '09). Association for Computing Machinery, New York, NY, USA, 2009. p. 117–132. DOI: 10.1145/1629575.1629587
12. **Vaarandi R.** A data clustering algorithm for mining patterns from event logs. Proc. 3rd IEEE Workshop IP Oper. Manage, Oct. 2003. p. 119–126. DOI: 10.1109/ipom.2003.1251233
13. **Du M., Li F., Zheng G., Srikumar V.** Deeplog: Anomaly detection and diagnosis from system logs through deep learning. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS), 2017. p. 1285-1298. DOI: 10.1145/3133956.3134015
14. **Zhang X., Li Z., Chen J.** Robust log-based anomaly detection on unstable log data. Proceedings of the 2019 27th ACM Joint Meeting, Tallinn, Estonia. 26–30 August 2019. p. 807–817. DOI: 10.1145/3338906.3338931
15. **Leemans S.J.J., Fahland D., van der Aalst W.M.P.** Discovering Block-Structured Process Models from Event Logs Containing Infrequent Behaviour. Business Process Management Workshops. Lecture Notes in Business Information Processing, vol 171. Springer, Cham, 2013. DOI: 10.1007/978-3-319-06257-0_6

16. **Le Q., Mikolov T.** Distributed Representations of Sentences and Documents. 31st International Conference on Machine Learning, ICML 2014, 2014. DOI: 10.18653/v1/s17-1003

Информация об авторе

Худяков Даниил Александрович, студент НГУ

Information about the Authors

Daniil A. Khudyakov, Student, Novosibirsk State University (Novosibirsk, Russian Federation)
ResearcherID: IAN-8563-2023

Статья поступила в редакцию 05.04.2023;
одобрена после рецензирования 19.04.2023; принята к публикации 19.04.2023
The article was submitted 05.04.2023;
approved after reviewing 19.04.2023; accepted for publication 19.04.2023

Правила оформления текста рукописи

Авторы представляют статьи на русском или английском языке объемом от 0,5 авторского листа (20 тыс. знаков) до 1 авторского листа (40 тыс. знаков), включая иллюстрации (1 иллюстрация форматом 190 × 270 мм = 1/6 авторского листа, или 6,7 тыс. знаков). Публикации, превышающие указанный объем, допускаются к рассмотрению только после индивидуального согласования с редакцией журнала.

Текст рукописи должен быть представлен в редколлегию в виде файла MS Word (.doc, .docx). Гарнитура Times New Roman, размер шрифта 11, межстрочный интервал 1, размеры полей – стандартные значения текстового редактора. Форматирование – выравнивание по ширине страницы, переносы слов включены, каждый новый абзац начинается с красной строки. Не допускается ручное форматирование абзацев (пробелами, лишними переводами строк, разрывами страниц).

Структура статьи

- Индекс УДК (универсальной десятичной классификации). Выравнивание по левому краю
- Название статьи. Выравнивание по центру, полужирный шрифт
- ФИО авторов (полностью). Выравнивание по центру, полужирный шрифт
- Места работы всех авторов. Выравнивание по центру, курсив
- Адреса электронной почты, ORCID авторов
- Аннотация статьи
- Ключевые слова, не более 10
- Благодарности, сведения о финансовой поддержке
- Название статьи **на английском языке**. Выравнивание по центру, полужирный шрифт
- ФИО авторов **на английском языке** (полностью). Выравнивание по центру, полужирный шрифт
- Места работы авторов **на английском язык**е. Выравнивание по центру, курсив
- Аннотация статьи **на английском языке (Abstract)**, 200–250 слов
- Ключевые слова **на английском языке (Keywords)**, не более 10
- Благодарности, сведения о финансовой поддержке **на английском язык**е, если есть соответствующий раздел на русском языке **(Acknowledgements)**
- Основной текст
- Список литературы / **References**
- Сведения об авторах

Требования к оформлению основного текста и иллюстративных материалов

Основной текст должен быть представлен в структурированном виде, рекомендуется использовать подзаголовки – например: Введение, Методика..., Выводы, Результаты, Заключение.

Подзаголовки отделяются и набираются полужирным шрифтом. В целях выделения частей текста и отдельных слов и словосочетаний допускается использование курсива или полужирного шрифта. Подчеркивание, разрядка, изменение основного кегля и выделение цветом не используются.

Иллюстрации к рукописи статьи должны быть приложены в виде отдельных файлов. При этом в тексте должно содержаться включенное изображение с указанием имени файла. Все иллюстрации, содержащие схемы, графики, алгоритмы и т. п., должны быть представлены в векторном виде (.ai, .eps, .cdr). Скриншоты и другие растровые изображения должны быть представлены в максимально высоком качестве, без каких-либо потерь и искажений (.jpg, .tif). Все иллюстрации должны иметь подрисуночную подпись – свое название. Надписи к таблицам и подписи к иллюстрациям приводятся **на двух языках (русском и английском)**.

Примеры:

Рис. 1. Диаграмма производительности...
Fig. 1. Performance diagram...

Таблица 1

Сравнение алгоритмов...

Table 1

Comparison of algorithms...

Нумерация последовательная и неразрывная от начала статьи. Не допускается использование других наименований, кроме «Рис.» / «Fig.», «Таблица» / «Table», и усложнение нумерации (например, «Рис. 3.2.»). Ссылка на иллюстрацию в тексте должна быть приведена в круглых скобках, например: (рис. 1), (табл. 1).

Формулы должны быть набраны с использованием редактора MathType либо встроенного редактора формул MS Word. Кегль основных символов – 11, греческие символы набираются прямым шрифтом, латинские – курсивом. Нумеруются только те формулы, на которые автор ссылается в тексте.

Abstract

Аннотация статьи на английском языке (Abstract) не должна быть дословным переводом русскоязычной аннотации. Раздел Abstract, как и основной текст, должен быть структурирован, в нем должно содержаться описание цели работы, методов исследования, научной значимости, выводов / результатов. Требуется качественный перевод на английский язык (при необходимости просим авторов обращаться к профессиональным переводчикам). **Объем Abstract 200–250 слов.**

Список литературы / References

Список литературы и список литературы на английском языке (References) размещаются в общем разделе. Рекомендуемое количество цитируемых в статье источников – не менее 10, в список желательно включать ссылки на актуальные работы по теме исследования, особенно в иностранных периодических изданиях.

В тексте статьи ссылки на литературу указываются цифрами в квадратных скобках, при необходимости указываются номера страниц, например: [2; 3. С. 15].

Список литературы нумеруется в порядке цитирования и оформляется в соответствии с ГОСТ Р 7.0.5-2008 на библиографическое описание (знаки тире в описании опускаются). Ссылки на неопубликованные работы, а также на Интернет-ресурсы (кроме электронных изданий, поддающихся библиографическому описанию) оформляются в виде сноски.

В Список литературы ссылки на источники следует включать на оригинальном языке опубликования. Каждый источник должен быть также оформлен на английском языке (References) по международному стандарту для публикаций в области информатики IEEE Style со следующими отличиями:

- инициалы авторов указываются после фамилии;
- название статьи не берется в кавычки, отделяется точкой;
- отсутствует союз «and» перед фамилией последнего автора;
- в диапазоне страниц – удвоенная «p» (например, «pp. 2–9»);

- год издания указывается после места издания (для книг) и сразу после названия журнала (для периодики).
- Перевод источника на английский язык:
- если источник имеет выходные данные на английском языке, то для формирования References **следует использовать именно эти данные**;
- если оригинальная публикация не содержит выходных данных на английском языке, то допускается транслитерация названия материала на латинский алфавит в сочетании с переводом на английский язык в квадратных скобках. В конце описания указывается, на каком языке написана эта работа, например, (in Russ.). При транслитерации можно воспользоваться Интернет-ресурсом http://ru.translit.ru/, рекомендуется выбрать стандарт BSI. Место издания не транслитерируется, указывается полностью на английском языке, например: Moscow. Название издательства / издателя, как правило, транслитерируется. Для журналов, у которых есть официальное название на английском языке, – использовать его (проверить на сайте журнала, или, например, в библиотеке WorldCat), если названия на английском языке нет, использовать транслитерацию по системе BSI. Не следует самостоятельно переводить названия журналов.

Если у цитируемого источника есть **цифровой идентификатор DOI** (https://search. crossref. org), его требуется обязательно указывать в конце библиографической ссылки.

Примеры оформления ссылок. Каждый источник в том же пункте дублируется на английском языке (References).

Источник на русском языке, перевод на английский доступен в метаданных статьи

1. **Журавлев С. С., Рудометов С. В., Окольнишников В. В., Шакиров С. Р.** Применение модельно-ориентированного проектирования к созданию АСУ ТП опасных промышленных объектов // Вестник НГУ. Серия: Информационные технологии. 2018. Т. 16, № 4. С. 56–67. DOI 10.25205/1818-7900-2018-16-4-56-67

Zhuravlev S. S., Rudometov S. V., Okolnishnikov V. V., Shakirov S. R. Model-Based Design Approach for Development Process Control Systems of Hazardous Industrial Facilities. *Vestnik NSU. Series: Information Technologies*, 2018, vol. 16, no. 4, pp. 56–67. (in Russ.) DOI 10.25205/ 1818-7900-2018-16-4-56-67

Источник на английском языке. Оформляем согласно требованиям для References. Приводим только 1 раз.

2. **Telnov V. I.** Optimization of the Beam Crossing Angle at the ILC for E + e- and yy Collisions. *Journal of Instrumentation*, 2018, vol. 13, no. 03, pp. P03020–P03020. DOI 10.1088/1748-0221/13/03/ p03020

Метаданные источника доступны только на русском языке

3. **Жижимов О. Л., Федотов А. М., Шокин Ю. И.** Технологическая платформа массовой интеграции гетерогенных данных // Вестник НГУ. Серия: Информационные технологии. 2013. Т. 11, вып. 1. С. 24–41.

Zhizhimov O. L., Fedotov A. M., Shokin Yu. I. Tekhnologicheskaya platforma massovoi integratsii geterogennykh dannykh [Technology Platform for the Mass Integration of Heterogeneous Data]. *Vestnik NSU. Series: Information Technologies*, 2013, vol. 11, no. 1, pp. 24–41. (in Russ.)

Сведения об авторах

Последний раздел статьи – информация об авторе / авторах **на русском и английском языках**:

- ФИО полностью, ученая степень, ученое звание;

- идентификаторы автора, такие как ResearcherID (всем авторам рекомендуется использовать данные сервисы для ведения актуального списка своих публикаций);
- контактный телефон (не публикуется).

Если статья представляется на английском языке, необходимо приложить перевод на русский язык названия, аннотации, ключевых слов, сведений об авторе.

Доставка материалов

Материалы предоставляются в редакцию по электронной почте inftech@vestnik.nsu.ru.

Порядок рецензирования

Все статьи сначала проходят проверку на заимствование и только после этого отправляются на рецензирование. Редакционный совет не допускает к публикации материал, если имеется достаточно оснований полагать, что он является плагиатом.

Тип рецензирования статей – двухуровневое, одностороннее анонимное («слепое»).

Для каждой статьи редколлегией выбираются рецензенты, научная деятельность которых связана с темой представленного материала. Ответственный секретарь журнала обращается к ним с просьбой дать экспертную оценку статье либо помочь организовать рецензирование.

Рецензии для журнала «Вестник НГУ. Серия: Информационные технологии» составляются по единой схеме и подразумевают оценку по следующим критериям: соответствие тематике журнала, оригинальность и значимость результатов, качество изложения материала.

Заполненный бланк рецензии высылается на электронный адрес редакции. В зависимости от экспертных заключений статья может быть принята редакционным советом к опубликованию, рекомендована автору к доработке (с последующим повторным рецензированием либо без него) или отклонена (с предоставлением автору мотивированного отказа). Автору на электронный адрес высылается текст рецензии без указания ФИО рецензента и его контактных данных.

Все рецензии хранятся в редакции журнала не менее 5 лет. Редколлегия журнала обязуется при поступлении соответствующего запроса направлять копии рецензий в Министерство науки и высшего образования Российской Федерации.