ПРОГНОЗИРОВАНИЕ ПРОФИЛЯ ФУНКЦИОНИРОВАНИЯ КОМПЬЮТЕРНОЙ СИСТЕМЫ НА ОСНОВЕ МНОГОЗНАЧНЫХ ЗАКОНОМЕРНОСТЕЙ

Шелухин О.И.[1], Раковский Д.И.[2]

Целью исследования *является разработка нового алгоритма прогнозирования состояний компьютерных систем (КС) на основе многозначных закономерностей (Алгоритм Прогноза Многозначных Зависимостей, АПМЗ). Состояния КС являются категориальными понятиями.*

Метод исследования: *анализ исторических данных с применением математического аппарата точечно-множественных (многозначных) закономерностей.*

Объектами исследования *являются теоретические и практические вопросы разработки, реализации и визуализации задач прогнозирования нормального и аномального профилей КС в целях информационной безопасности.*

Результат исследования. *Разработаны методология и алгоритм прогнозирования состояния компьютерных систем АПМЗ. Обоснованы границы изменения входных параметров алгоритма, которые необходимо настроить для корректной выдачи прогноза.*

Разработана программная реализация предложенного алгоритма прогнозирования АПМЗ. Работоспособность алгоритма протестирована на реальных экспериментальных данных. Проведен пространственный анализ результатов прогнозирования.

Научная значимость: *расширена область применения математического аппарата точечно-множественных закономерностей; предложен новый алгоритм прогнозирования нормального и аномальных состояний КС, являющихся категориальными понятиями. Разработанный алгоритм прогнозирования АПМЗ может быть обобщен на иные предметные области, содержащие исторические данные.*

Ключевые слова: *исторические данные, анализ временных рядов, модель прогнозирования, прогнозирование временных рядов, компьютерная система, прогноз аномалий, состояние системы.*

DOI:10.21681/2311-3456-2022-6-53-70

I. Введение

Системы обнаружения атак, основанные на методах обнаружения аномалий, демонстрируют высокие показатели эффективности обнаружения атак как известного, так и не известного типов [1, 2]. Принцип работы методов обнаружения аномалий основан на автоматическом построении модели нормального поведения КС на основании данных о ее функционировании «в прошлом». Актуальной является задача прогнозирования профиля функционирования КС с целью своевременного обнаружения отклонения от нормального поведения. Решение данной задачи сводится к прогнозированию временных рядов на основании «исторических данных» [3, 4], учитывающих поведение системы «в прошлом» [5—7].

Прогнозирование временных рядов с применением математического аппарата статистического анализа основано на однозначных закономерностях вида: $f : X \to Y$ или $f(x) = y$. Использование подобных закономерностей требует ввода случайной компоненты (погрешности измерения) с определенными характеристиками, характеризующими разброс погрешностей некоторым законом распределения. Использование статистического анализа требует принятия ряда гипотез, проверка которых обычно является сложной задачей [8].

Наиболее перспективными алгоритмами в настоящее время является семейство структурных моделей прогнозирования [9], к которым относят модели на ос-

1 Шелухин Олег Иванович, доктор технических наук, профессор, заведующий кафедрой «Информационная безопасность» (Московский технический университет связи и информатики), Москва, Россия. E-mail: sheluhin@mail.ru

2 Раковский Дмитрий Игоревич, ассистент кафедры «Информационная безопасность» (Московский технический университет связи и информатики), Москва, Россия. E-mail: Prophet_alpha@mail.ru

нове методов и алгоритмов интеллектуального анализа данных: методы опорных векторов [10]; методы, основанные на кластеризации, расстоянии и плотности [11]; комплексные методы [12, 13], нейронные сети [14, 15].

В отличие от известных, целью исследования является разработка нового алгоритма прогнозирования состояний КС (нормальных и аномальных), являющихся категориальными понятиями. Разработанный в работе алгоритм, названный «Алгоритм Прогноза Многозначных Зависимостей» (АПМЗ) основан на анализе исторических данных с применением математического аппарата точечно-множественных (многозначных) закономерностей [16, 17].

II. Формализация алгоритма АПМЗ

Введем в рассмотрение множество *A*, состоящее из *m* наборов значений дискретно изменяющихся атрибутов: $A \subseteq \{A_1 \times A_2 \times \ldots \times A_m\}$ наблюдаемых "в прошлом", каждый из которых содержит *n* элементов (*n* равно количеству наблюдений, записей «исторических данных»).

Для оценки значений, характеризующих уровень обслуживания КС (Service Level Objectives, SLO, [18]) на основе «исторических данных» по каждому атрибуту, введем в рассмотрение ряд из *k* показателей (состояний) КС: $S = \{s_1, s_2, \ldots, s_k\}$. Число состояний определяется задаваемыми порогами $P = \{p_1, p_2, \ldots, p_{k-1}\}$ SLO, превышение которых приводит к изменению функционирования КС. Дополнительно вводится понятие «нормального» состояния КС — когда нарушений SLO не происходит. Состояние КС в каждый момент времени *t* характеризуется категориальным маркером $label_t$ и совокупной оценкой значений соответствующих атрибутов системы по порогам SLO: $label_t \subset S$. В дальнейшем рассматривается случай, когда каждый порог p_j однозначно соотносится с соответствующими значениями атрибутов A_i: $|P| = |A|, k-1 = m$.

При одновременном превышении нескольких порогов *P* SLO, КС может находиться одновременно в нескольких состояниях (состояниях нарушения по нескольким SLO; $|label_t| > 1$).

В результате «исторические данные» о поведении КС можно представить в виде:

$$D_n = \{(a_1, s_1), \ldots, (a_n, s_n) | (a_i, s_i) \in S \times A\}. \quad (1)$$

Поскольку состояние КС однозначно определяется множеством атрибутов *A*, то в дальнейшем задача прогнозирования состояния на основании «исторических данных» основывается только на введенных в рассмотрение состояниях КС — S. В результате соотношение (1) может быть преобразовано в одномерную последовательность маркеров $label_t$, характеризующих состояния КС в каждый текущий момент времени *t*:

$$Label = \{label_t\}; t = \overline{1, n}. \quad (2)$$

Под последовательностью маркеров будем понимать ограниченный и упорядоченный во времени набор маркеров состояний КС.

Введем в рассмотрение понятие «зависимость», под которой будем понимать мультимножество троек : «последовательность размера *τ* – сдвиг *step* – соответствующее состояние КС», образованных из последовательности маркеров $label_t$ (2) по правилу:

$$Lib(step, \tau) = \begin{cases} L_{cou} F_{cou,step} | L_{cou} = \\ = \{Label(cou + j)\} \end{cases}$$
$$cou = \overline{1, n - step - \tau}, j = \overline{0, \tau - 1} \quad (3)$$

Здесь L_{cou} – последовательность маркеров состояния *КС Label* (2) с индексами от *cou* до *cou*+*τ*-1; *cou* – сдвиг последовательности; τ – размер фрагмента последовательности; *j* – счетный индекс для формирования последовательности маркеров состояния КС; step – сдвиг «в будущее» относительно последовательности маркеров L_{cou}; $F_{cou,step} : L_{cou} \to S$ – состояние КС, соответствующее последовательности L_{cou}, и «сдвинутое» относительно нее на step значений «в будущее».

Запись (3) является многозначной [19] зависимостью. В результате задача прогнозирования состояния КС сводится к выбору последовательности в зависимости (3), которая наилучшим образом описывает прогнозируемое состояния КС «в будущем».

В результате многозначная зависимость для такого набора данных представляет собой запись следующего вида:

$$A \times SI\mu, \quad (4)$$

где $\mu : A(I) \times S \to N \bigcup \{0\}$; $A(I) = \prod_{i \in I} A_i$; N – множество натуральных чисел.

Параметр *μ*, показывает, сколько раз пара вида «значения атрибутов – состояние КС» $\mu(a, s)$ встречалась в выборке. Назовем *μ* числом комбинаций.

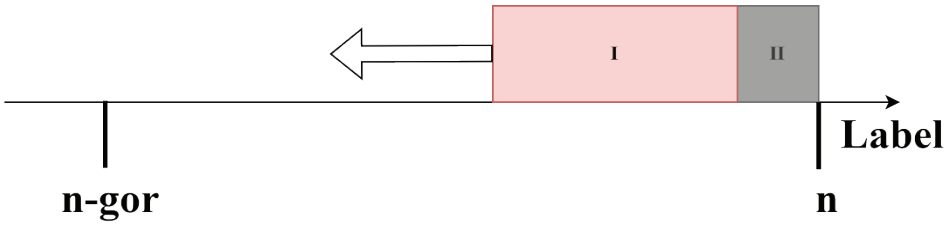

Рис. 1. Процесс формирования зависимости из «исторических данных» в пределах [n-gor; n]

III. Структура АПМЗ

Зададимся горизонтом прогнозирования *gor*, под которым будем понимать количество «прошлых» записей в наборе маркеров состояний КС (2) используемых для построения прогноза. Для прогнозирования зададим ограниченный набор размеров фрагментов последовательности *τ*, который будет определять диапазон изменения размера масок (непустого подмножество в множестве полей аргументов фрагмента последовательности *τ*):

$$Mask = \{mask\}, mask = \overline{minLength, maxLength}, \quad (5)$$
$$minLength > 0, maxLength < gor - step.$$

где *minLength* – минимальный размер маски; *maxLength* — максимальный размер маски; *gor* – горизонт; *step* — точка, которую необходимо спрогнозировать.

Для устранения ситуаций, когда в силу неопределенности итоговый прогноз не может быть выдан однозначно, вводится минимальный порог голосования — *votelevel*.

При формировании прогноза будем основываться на выборе наилучшего по надежности состояния КС, вычисляемого по таблицам статистик экспериментальных данных, сформированных на основе заданного диапазона размерности масок *Mask* в пределах выбранного горизонта *gor*. С этой целью для формирования прогноза в пределах промежутка [*n-gor*; *n*] необходимо преобразовать (1) к виду последовательности маркеров (2). При этом запись под индексом *n* считается наиболее актуальной записью «исторических данных», соответствующей текущему моменту времени.

На основе введенных понятий АПМЗ может быть представлен в виде последовательности этапов.

IV. Этапы прогнозирования состояния КС

A. Этап 1. Формирование набора масок

Исходя из размера фрагмента последовательности *τ*, формируется множество масок $MASK_\tau$, мощностью $2^\tau - 1$. Каждый элемент такого множества, являясь непустым подмножеством в множестве полей аргументов фрагмента последовательности *τ*, представляет из себя булевый вектор, сформированный по правилу:

$$MASK_\tau = \left\{mask_{(2)}, mask = \overline{1, 2^\tau}\right\}, \quad (6)$$

где $mask_{(2)}$ – двоичное представление числа $mask$ из диапазона $1, 2^\tau$.

Данная операция выполняется для всего диапазона изменения размера масок (5). Итогом данного этапа является библиотека масок:

$$MaskLibrary = \left\{MASK_t, \tau = \overline{minLength, maxLength}\right\}.$$

Каждый элемент такой библиотеки, в соответствии с (6), содержит $2^\tau - 1$ элементов.

B. Этап 2. Формирование набора шаблонов

Для каждого размера фрагмента последовательности *τ* из «исторических данных» [*n-gor*; *n*] формируются последовательности (2), и на их основании формируется зависимость (3).

Иллюстрация этого процесса (формирование троек «последовательность размера *τ* – сдвиг *step* – соответствующее состояние КС») из «исторических данных» в пределах [*n-gor*; *n*] приведена на рис. 1.

Каждому множеству масок MASKт сопоставляется соответствующая зависимость $Lib(step, \tau)$, после чего из данной пары формируется шаблон путем поэлементного перемножения последовательности *L* (одномерной последовательности зависимости (3)) на маску $mask_{(2)}$ – элемент множества масок — (наложение маски на последовательность) по следующему правилу:

$$conk\left(L, mask_{(2)}\right) = L \dot{\&} mask, \quad (7)$$

где $\dot{\&}$ — модифицированная операция поэлементной конъюнкции, $mask_{(2)}$ – элемент множества масок (6) в двоичном виде, а *L* – одномерная последовательность зависимости (3). При этом мощность маски и последовательности должны быть одинаковыми ($|L| = |mask|$).

Таблица 1

Таблица истинности для модифицированной операции поэлементной конъюнкции example-ного разряда маски на example-ный элемент множества L

L(example)	$mask_{(2)}$ **(example)**	$L \dot{\&} mask$
L(example)	0	zero
L(example)	1	*L(example)*

Введение операции $\dot{\&}$ обосновано двумя причинами:

1) в качестве «разрядов» в операции (7) выступают элементы множеств *L* и разряды маски $mask_{(2)}$;

2) результат выполнения такой операции отличен от стандартной поразрядной конъюнкции (см. таблицу 1 истинности, в которой приводится пример наложения *example*-ного разряда маски на *example*-ный элемент множества *L*).

Результат конъюнкции элемента множества *L* и нулевого разряда маски *mask* (zero) отличен от «0» и его свойство будет рассмотрено на этапе 3.

В соответствии с правилом (7), формирование шаблона для всей зависимости (3) для конкретной маски $mask_{(2)}$ из (6) будет выглядеть следующим образом. Для каждой последовательности L_{cou} формируется соответствующий шаблон для конкретной маски $mask_{(2)}$ по правилу:

$$formprototemp\left(L_{cou}, mask_{(2)}\right) = \{(conk\left(L_{cou}, mask_{(2)}\right)\}.$$

Соответствующий маркер состояния КС $F_{cou,step}$ сопоставляется получившемуся шаблону без изменений:

$$temp\left(L_{cou}, mask_{(2)}, F_{cou,step}\right) = \\ = \left(formprototemp\left(L_{cou}, mask_{(2)}\right), F_{cou,step}\right). \quad (8)$$

Тогда набор шаблонов для всей зависимости (3) (набор шаблонов, соответствующий маске $mask_{(2)}$, (8)) будет иметь вид для конкретной маски $mask_{(2)}$:

$$template_{mask_{(2)}} = \left\{temp\left(L_{cou}, mask_{(2)}, F_{cou,step}\right)\right\},$$
$$\overline{cou = 1, n - step - \tau}.$$

Библиотеку сформированных шаблонов для всего множества масок (6) запишем как:

$$TemplateLib_{\tau} = \left\{template_{mask_{(2)}}, \overline{mask_{(2)} = 1, 2^{\tau}}\right\}. \quad (9)$$

Тогда общая библиотека шаблонов (9) для всех фрагментов последовательности (7) по всей длине макси (5) будет иметь вид:

$$TemplateLibrary = \\ = \left\{TemplateLib_t, \overline{\tau = minLength, maxLength}\right\}. \quad (10)$$

Параметры *minLength, maxLength* задаются в соответствии с введенными ограничениями, описанными в (5).

С. Этап 3. Отбор шаблонов, соответствующих настоящему моменту времени

Из полученной библиотеки шаблонов (10) необходимо отобрать элементы, соответствующие настоящему моменту времени. Под текущим моментом времени понимается последовательность вида: $\overline{presentLabel_{\tau} = \left\{Label\left(time\right)\right\}, time = n - \tau, n}$.

Отбор производится по каждой маске $\overline{mask_{(2)} = 1, 2^{\tau}}$, соответствующей шаблону $TemplateLib_{\tau}$ по каждому $\tau = \overline{minLength, maxLength}$. Путем поиска совпадения шкал аргументов между шаблонами $TemplateLib_{\tau}$ и последовательностью $presentLabel_{\tau}$ отбираются двойки $\left(cou, template_{mask_{(2)}, iter}\right)$ вида «номер шаблона — шаблон» по следующему правилу:

$$SelTemplateLabels_{\tau, mask_{(2)}} = \\ = \left\{\begin{array}{c}\left(cou, TemplateLib_{\tau, iter}\right) | \; TemplateLib_{\tau, iter} \\ \cong presentLabel_{\tau} \\ \overline{iter = 1, |TemplateLib_{\tau}|}\end{array}\right. \quad (11)$$

где $TemplateLib_{\tau, iter}\left(1\right)$ – первый элемент $template_{mask_{(2)}}$, из которых состоит библиотека сформированных шаблонов $TemplateLib_{\tau}$, состоящая из последовательности L_{cou} с наложенной на нее маской $mask_{(2)}$.

Под знаком $\cong$ понимается совпадение шкал аргументов шаблона и последовательности текущего времени по следующему правилу:

$$rule(tElem, pLElem) = \begin{cases} 1, tElem = pLElem \\ 1, tElem = zero \\ 0, \text{иначе} \end{cases},$$

где $tElem$ – элемент последовательности $formprototemp\left(L_{cou}, mask_{(2)}\right)$ в двойке $template_{mask_{(2)},cou}$; $pLElem$ – элемент последовательности текущего времени. «zero» — результат поэлементной конъюнкции (6). Данное значение может быть интерпретировано как «любой элемент на этом месте». Операция (11) выполняется для каждого элемента библиотеки шаблонов (10):

$$SelTemplateLibrary = \\ = \{SelTemplateLabels_{\tau,mask_{(2)}}\}, \\ mask_{(2)} = \overline{1, 2^{\tau}}, \tau = \overline{minLength, maxLength}$$

Результатом этапа 3 является библиотека отобранных шаблонов (набор шаблонов, соответствующих текущему моменту времени).

D. Этап 4. Формирование взвешенных шаблонов

Каждая пара $\left(cou, template_{mask_{(2)},iter}\right)$, отобранная по правилу (11), состоит из: временной метки cou, соответствующей данной последовательности и шаблона $template_{mask_{(2)},iter}$, состоящего, в свою очередь, из пары $\left(formprototemp\left(L_{cou}, mask_{(2)}\right), F_{cou,step}\right)$.

В этой паре $formprototemp\left(L_{cou}, mask_{(2)}\right)$ – последовательность L_{cou} с наложенной на нее маской $mask_{(2)}$, $F_{cou,step}$ соответствующий маркер состояния КС с учетом сдвига на *step* шагов.

Зная номер последовательности cou, каждому шаблону возможно сопоставить весовой коэффициент $W\left(cou\right)$, зависящий от удаленности шаблона от текущего времени в виде:

$$W\left(cou\right) = \delta^{n-cou}, \quad (12)$$

где n – число, соответствующее количеству записей в «исторических данных» D_n и характеризующее текущий момент времени внутри алгоритма.

Параметр δ, 0 < δ ≤ 1 характеризует влияние удаленности шаблона на результат прогноза. Если он равен 1, то все шаблоны оказывают одинаковое влияние, независимо от того как далеко в прошлом они были обнаружены.

Сформируем из пары «номер последовательности — шаблон» $cou, template_{mask_{(2)},iter} \in SelTemplateLabels_{\tau,mask_{(2)}}$ и соответствующего ей весового коэффициента (12) тройку $cou, F_{cou,step}, W\left(cou\right)$ по правилу:

$$stat_{mask_{(2)}} = \left\{ \begin{matrix} \{cou, F_{cou,step}, W\left(cou\right)\} \\ \forall SelTemplateLabels_{\tau,mask_{(2)}} \end{matrix} \right\}, \quad (13)$$

где $F_{cou,step}$ является значением маркера состояния КС в $template_{mask_{(2)}}$.

Назовем тройку $cou, F_{cou,step}, W\left(cou\right)$ взвешенным шаблоном, а (13) – набором взвешенных шаблонов.

Тогда для всего перечня шаблонов $SelTemplateLabels_{\tau}$ набор взвешенных шаблонов примет вид:

$$weightLib_{\tau} = \{stat_{mask_{(2)}}\} \\ \forall template_{mask_{(2)}} \in SelTemplateLabels_{\tau}, \\ mask_{(2)} = \overline{1, 2^{\tau}}. \quad (14)$$

Назовем (14) библиотекой взвешенных шаблонов. Сформируем библиотеку отобранных взвешенных шаблонов для всего диапазона τ:

$$WeightLibrary = \{weightLib_{\tau}\}, \\ \tau = \overline{minLength, maxLength}. \quad (15)$$

E. Этап 5. Группировка взвешенных шаблонов

К началу этапа 5 имеется набор отобранных взвешенных шаблонов $WeightSTLibrary$, состоящий из наборов взвешенных шаблонов для каждой маски $weightLib_{\tau}$ с разным количеством элементов внутри. В случае достаточно большого диапазона $\tau = minLength, maxLength$ в совокупности с большим горизонтом *gor* общее количество взвешенных шаблонов может накладывать существенные ограничения на точность и/или время, затраченное на получение прогноза.

При этом часть маркеров состояний КС может быть не представлена в наборе взвешенных шаблонов $stat_{mask_{(2)}}$, либо представлена недостаточным количеством записей для участия в последующем прогнозировании. С целью учета «представленности» маркера состояния КС к взвешенного шаблона по маске $stat_{mask_{(2)}}$ добавим столбец Y, характеризующий удаленность шаблона в $stat_{mask_{(2)}}$ от текущего момента времени:

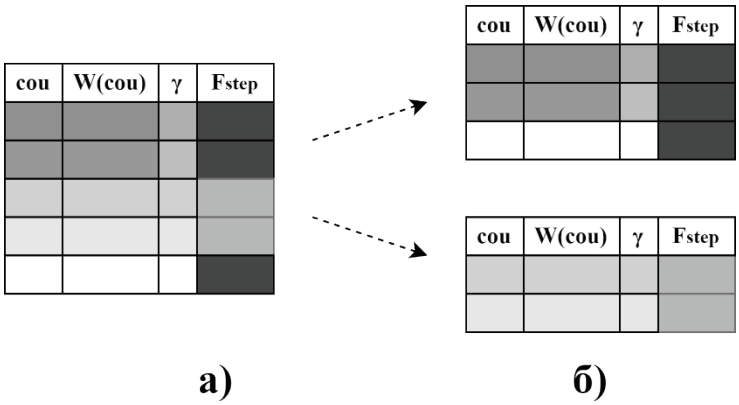

Рис. 2. Процесс группировки взвешенных шаблонов: до (а) и после (б)

$$mstat_{mask_{(2)}} = stat_{mask_{(2)}} Y,$$
$$Y = \left\{ \gamma^{\left|stat_{mask_{(2)}}\right| - z}, \; z = \overline{1, \left|stat_{mask_{(2)}}\right|} \right\}. \quad (16)$$

Таким образом, каждый взвешенный шаблон дополняется коэффициентом γ, учитывающим актуальность записей внутри набора взвешенных шаблонов. Разница между δ и γ заключается в разной мощности библиотек шаблонов до отбора по соответствию настоящему времени и после.

Модифицируем библиотеку взвешенных шаблонов (15) и библиотеку отобранных шаблонов (16).

$$mweightLib_\tau = \left\{ mstat\left(template_{mask_{(2)}}\right) \right\},$$
$$mask_{(2)} = \overline{1, 2^\tau},$$
$$template_{mask_{(2)}} \in SelTemplateLabels_\tau.$$

Поскольку сущность (16) не отличается от (13), за исключением добавления нового столбца, далее будем понимать под $mstat_{mask_{(2)}}$ – набор взвешенных шаблонов, а элементы $mstat_{mask_{(2)}}$ – взвешенные шаблоны.

Отбор будем производить следующим образом. Разделим каждый набор взвешенных шаблонов $mstat_{mask_{(2)}}$ на k подмножеств по количеству состояний КС S: $mstat_{mask_{(2)}} = Group_{s_1} \cup Group_{s_2} \cup \ldots \cup Group_{s_k}$.

Элементы каждого подмножества $Group_s \subseteq mstat_{mask_{(2)}}$, содержат взвешенные шаблоны, относящиеся только к определенному состоянию КС ($\forall gr \in Group_s, gr = s$). Сформируем множество $grouptmpl_{mask_{(2)}}$, содержащее все подмножества:

$$grouptmpl_{mask_{(2)}} = \left\{ Group_{s_1} \cup Group_{s_2} \cup \ldots \cup Group_{s_k} \right\}. \quad (17)$$

Назовем подмножество $Group_s$ – набором взвешенных шаблонов по состоянию S, а $grouptmpl_{mask_{(2)}}$ – набором сгруппированных взвешенных шаблонов. Выполним группировку (17) для каждой маски:

$$grouptmplLib_\tau = \left\{ grouptmpl_{mask_{(2)}}, mask_{(2)} = \overline{1, 2^\tau} \right\}. \quad (18)$$

После — для каждого фрагмента последовательности (18):

$$GroupedLib = \left\{ grouptmplLib_\tau \right\}, \quad \tau = \overline{minLength, maxLength}. \quad (19)$$

Иллюстрация процесса группировки взвешенных шаблонов приведена на рис. 2. На иллюстрации приведен упрощенный вид взвешенного шаблона (рис. 2а), где вместо значений приведены оттенки серого. На иллюстрации показано: зависимость функции *W(cou)* от *cou*; характер различия между значениями *W(cou)* и γ. Категориальные значения F_{step} также показаны разными оттенками серого цвета. Сгруппированные взвешенные шаблоны по значению F_{step} (состоянию S, $Group_s$) приведены на рис. 2б.

Для каждого набора взвешенных шаблонов по состоянию S $Group_s$ вычислим надежность *reliability*, характеризующий «вес» соответствующего состояния КС в наборе сгруппированных взвешенных шаблонов:

$$reliability\left(Group_s\right)=\sum_{Y\in Group_s} \mathrm{Y}. \quad (20)$$

Для всех элементов (18) для каждого фрагмента последовательности (19) вычислим надежность по правилу (20), после чего проведем отбор. Отбор будем проводить по следующему правилу:

$$relrule\left(Group_S, stateLevel(\tau)\right)=$$
$$=\begin{cases}1, reliability(Group_S)\geq stateLevel(\tau)\\ 0, \text{иначе}\end{cases}$$

где *state_level(τ)* — порог веса взвешенного шаблона, зависящий от размера маски.

Не прошедшие отбор состояния КС исключаются из последующего процесса прогнозирования. Множество из прошедших отбор набора сгруппированных взвешенных шаблонов $grouptmpl_{mask_{(2)}}$ по порогу веса взвешенного шаблона *state_level(τ)* обозначим как:

$$Selgrouptmpl_{mask_{(2)}}=$$
$$=\left\{\begin{matrix}\left\{Group_s\right\}\mid relrule\left(Group_s, stateLevel\left(\tau\right)\right)=1\\ Group_s\in grouptmpl_{mask_{(2)}}\end{matrix}\right\}. \quad (21)$$

Назовем (21) сгруппированными взвешенными шаблонами, прошедшими отбор по весу. Сформировав набор сгруппированных взвешенных шаблонов, прошедших отбор по весу, получим библиотеку отобранных по порогу веса взвешенных шаблонов для каждой маски:

$$SelgroupedtempleLib_\tau=$$
$$=\left\{Selgrouptmpl_{mask_{(2)}}, mask_{(2)}=\overline{1,2^\tau}\right\}.$$

А также библиотеку отобранных по порогу веса взвешенных шаблонов для каждого фрагмента последовательности:

$$GroupedLib=\left\{SelgroupedtempleLib_t\right\},$$
$$\tau=\overline{minLength, maxLength}. \quad (22)$$

F. Этап 6. Отбор взвешенных шаблонов

На данном этапе в каждом наборе сгруппированных взвешенных шаблонов, прошедших отбор по весу $Selgrouptmpl_{mask_{(2)}}$ (21) отобраны группы взвешенных шаблонов, имеющие надежность *reliability* большую, чем *state_level(τ)*.

При большом горизонте *gor* большинство таблиц содержат неактуальные записи. С целью отсечения неактуальных последовательностей, произведем отбор по порогу надежности *reliabilityLevel* среди взвешенных шаблонов в каждом шаблоне $Selgrouptmpl_{mask_{(2)}}$. Для этого упростим строение набора взвешенных шаблонов по состоянию S $Group_s\in Selgrouptmpl_{mask_{(2)}}$, исключив из него столбцы с итерационной переменными cou, $Y\left(iter\right)$, поскольку они более не потребуются. Также, поскольку состояние КС в наборе $Group_s$, не меняется (все $F_{cou,step}=s$)), данный столбец также можно опустить. Упрощенный набор взвешенных шаблонов по состоянию S запишем так:

$$primeGroup_s=\left\{W\left(cou\right)\right\}, \forall cou\in Group_s.$$

Отбор будет производиться по следующему правилу: если $\sum_{primeGroup_s} W\left(cou\right)>reliabilityLevel$ – то набор взвешенных шаблонов по состоянию S участвует в прогнозировании, иначе – отбрасывается.

Набор сгруппированных взвешенных шаблонов, в котором остались только отобранные по весу наборы взвешенных шаблонов, запишем как пару «состояние КС – суммарная надежность»:

$$SelprimeGroup_{mask_{(2)}}=$$
$$=\left\{S, \sum_{\substack{primeGroup_s\\ \forall cou\in Group_s}} W\left(cou\right)\right\}\forall Group_s \quad (23)$$
$$\in grouptmpl_{mask_{(2)}},$$

Следовательно, упрощенное множество из прошедших отбор набора сгруппированных взвешенных шаблонов $grouptmpl_{mask_{(2)}}$ по порогу веса взвешенного шаблона *state_level(τ)* (21), запишем как:

$$SelprimeGroupLib_\tau=$$
$$=\left\{SelprimeGroup_{mask_{(2)}}, mask_{(2)}=\overline{1,2^\tau}\right\},$$

а упрощенную библиотеку отобранных по порогу веса взвешенных шаблонов для каждого фрагмента последовательности (22) запишем как:

$$SelGroupedLib=\left\{SelprimeGroupLib_\tau\right\},$$
$$\tau=\overline{minLength, maxLength}.$$

Результатом данного этапа является набор пар набор пар вида «состояние КС – надежность» полученный для каждого оставшегося набора сгруппированных взвешенных шаблонов по каждой маске для каждого фрагмента последовательности *т*.

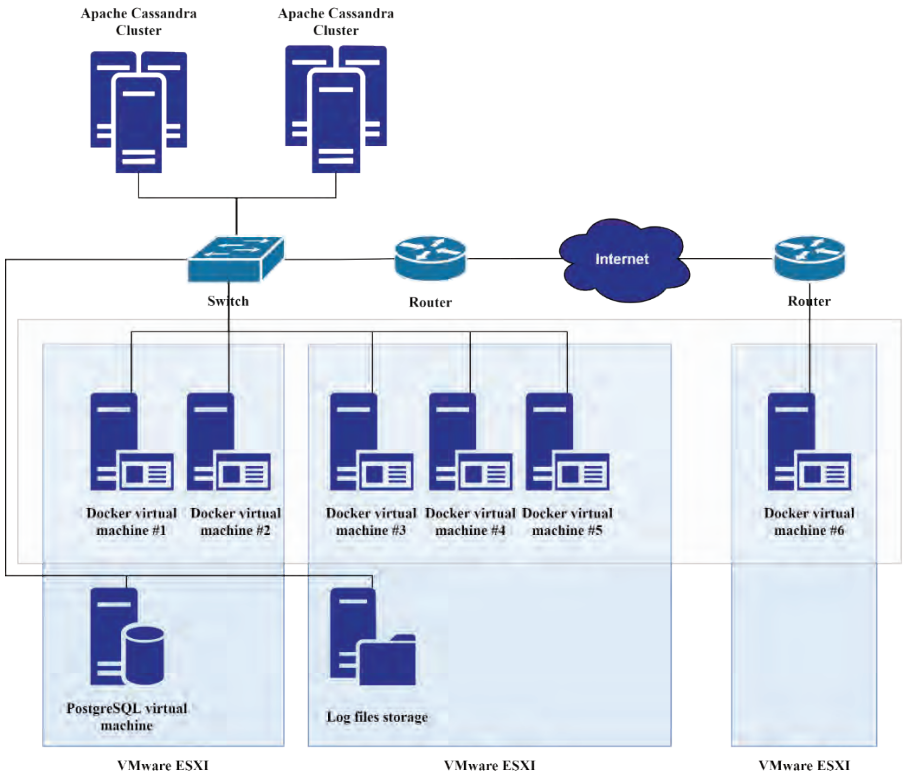

Рис. 3. Схема исследуемой сетевой инфраструктуры

G. Этап 7. Голосование о выборе состояния КС

По каждому набору пар вида «состояние КС –надежность» (23) происходит голосование о выборе состояния КС. Пары $\left(S, \sum_{\substack{primeGroups_S \\ \forall cou \in Groups_S}} W(cou)\right)$ упорядочиваются по убыванию суммы весов, после чего происходит процедура взвешивания и голосования по следующему правилу:

$$\frac{SelprimeGroup_{mask_{(2)}}(1) - SelprimeGroup_{mask_{(2)}}(2)}{\sum SelprimeGroup_{mask_{(2)}}} \geq \geq wdifference .$$

Если порог голосования $wDifference$ превышен, то маркер состояния КС с наибольшим весом считается победившим в голосовании.

Данная операция выполняется для каждого оставшегося взвешенного шаблона по каждой маске $mask_{(2)} = 1, 2^{\tau}$ для каждого фрагмента последовательности τ. Набор полученных в результате голосования состояний КС может быть сведен к одномерному перечню состояний КС $s \in S$.

$$voteLibrary = \left\{vote_1, vote_2, \ldots vote_{|GroupedLib|}\right\}.$$

H. Этап 8. Подсчет голосов

В полученном наборе голосов $voteLibrary$ ведется подсчет количества голосов по каждому состоянию КС. Итог подсчета голосов заносится в массив:

$$CountLibrary = \left\{Count_{s1}, \ldots, Count_{sk}\right\}$$

I. Этап 9. Выдача прогноза

Производится финальная проверка набора голосов (24) на соответствие условию минимального порога голосования для выдачи прогноза. Для этого по аналогии с этапом 7, набор голосов (24) ранжируется по убыванию. Если:

$$\frac{CountLibrary(1) - CountLibrary(2)}{\sum CountLibrary} \geq votelevel ,$$

то полученные веса считаются удовлетворяющими условию минимального порога голосования для выдачи прогноза. В противном случае происходит отказ от прогнозирования при заданных входных параметрах (*state_level(τ), reliability_level, wdifference, votelevel.* Итоговым прогнозом является состояние КС, имеющее максимальный вес в наборе весов (24):

$$Prognosus = \max\left[CountLibrary\right]$$

Итогом прогнозирования является состояние КС, выбранное при помощи соотношения (25).

Таблица 2

Пример единичной записи исходных экспериментальных данных

№	**ts_epoch,cpu_idle,cpu_iowait,cpu_nice,cpu_softirq,cpu_system,cpu_total,cpu_user,memory_actualfree,** **memory_actualusedbytes,memory_actualusedpct,memory_free,memory_swapfree,memory_swapusedbytes,memory_swapusedpct,disk_await,disk_busy,disk_ioreadawait,disk_ioreadmergespersec,** **disk_ioreadrequestspersec,disk_iostatqueueavgsize,disk_iostatrequestavgsize,disk_iowriteawait,** **disk_iowritemergespersec,disk_iowriterequestspersec,disk_readbytes,disk_readcount,disk_readtime,** **disk_writebytes,disk_writecount,disk_writetime,load_fifteenminutes,load_fiveminutes,load_oneminute,** **network_inbytes,network_inpackets,network_outbytes,network_outdropped,network_outpackets,** **dns_answerscount,dns_networkbytes,dns_srcport,http_dstport,http_networkbytes,http_requestbytes,** **http_responsebytes,http_srcport,ping_avg,ping_max,ping_min,server_response_timeconnect,** **server_response_timenamelookup,server_response_timepretransfer,server_response_timestarttransfer,** **server_response_timetotal,timestamp**
	...
153	1569303459,6.19295,0.01363333333333333,0.0015166666666666668,0.01806666666666666,0.37335,1.7934166666666669,1.40045,42750029824.0,24453439488.0,0.3638666666666666,8379613866.666667,33687597056.0,1048576.0,0.0,0.6,0.7,0.0,0.0,0.0,0.0,4770.6,0.6,0.0,12.1,29285.2,1236894105.6,46565.8,6235915.666666668,128192291328.0,59254311.96666667,2.13,2.02,1.76,759439514.0,1118806.5,735361597.0,0.0,1048070.0,0.0,58.0,36591.0,8080.0,257.0,120.0,137.0,56928.0,2.603,2.777,2.385,0.005,0.004,0.005,0.959,0.96,2019-09-24 05:37:39
	...

V. Апробация АПМЗ. Описание экспериментальных данных

Для апробации АПМЗ в качестве исходных взяты были экспериментальные данные функционирования КС, состоящей из 6 хостов, образующих кластер под управлением Rancher [20] (рис. 3).

В ходе эксперимента были получены измерения с шагом 1 секунда, содержащие себе 57 числовых атрибутов и 10 категориальных атрибутов. Экспериментальные данные представлены в виде таблицы с количеством записей, равным 237835.

В качестве иллюстрации экспериментальных данных рассмотрим пример, представленный в таблице 2. Для компактности, все атрибуты, перечисленные через запятую, за исключением номера записи, собраны в правом столбце. В представленной записи значения, характеризующие КС по каждому атрибуту, также приведены через запятую.

Как показано в работе [21], целесообразно исключить из анализа атрибуты категориального типа, а также атрибуты, не изменяющиеся во времени. Результирующий перечень анализируемых атрибутов КС метрического типа и их идентификаторы представлены в табл. 3. Курсивом помечены атрибуты, имеющие постоянное значение на протяжении всего эксперимента. В дальнейших экспериментах данные атрибуты не задействовались.

Проведенный в [22] дополнительный отбор атрибутов показал, что их количество можно сократить до 34 шт. Последующее сокращение количества атрибутов по информационной значимости позволило получить набор из 4 атрибутов.

Оставшиеся атрибуты содержат информацию о дисковых операциях в КС (*disk_ioreadmergespersec*) и о сетевом траффике (*server_response_timetotal, ping_avg*). В перечень анализируемых атрибутов также включена информация об отброшенных сетевых пакетах (*network_outdropped*). Показатели уровня обслуживания и соответствующие им состояния КС сформированы в виде порогов, определяющих категориальные маркеры состояния КС:

Таблица 3

Перечень анализируемых атрибутов КС метрического типа

№	Тип атрибута	Идентификатор атрибута
1	**Данные по использованию центрального процессора**	idle(1), iowait(2), *irq*, nice(3), softirq(4), *steal*, system(5), total(6), user(7)
2	**Данные по использованию памяти**	actualfree(8), actualusedbytes(9), actualusedpct(10), free(11), swapfree(12), *swaptotal*, swapusedbytes(13), swapusedpct(14), *total*
3	**Данные по использованию диска**	await(15), busy(16), ioreadawait(17), ioreadmergespersec(18), ioreadrequestspersec(19), iostatqueueavgsize(20), iostatrequestavgsize(21), iowriteawait(22), iowritemergespersec(23), iowriterequestspersec(24), readbytes(25), readcount(26), readtime(27), writebytes(28), writecount(29), writetime(30)
4	**Данные по средней загрузке центрального процессора**	oneminute(31), fiveminutes(32), fifteenminutes(33)
5	**Данные по использованию сети**	inbytes(34), *indropped, inerrors*, inpackets(35), outbytes(36), outdropped(37), *outerrors*, outpackets(38)
6	**Данные по DNS запросам**	networkbytes(39), answerscount(40)
7	**Данные по HTTP запросам**	networkbytes(41), requestbytes(42), responsebytes(43)
8	**Данные PING запроса для определения задержки сигнала в сети**	avg(44), max(45), min(46)
9	**Данные CURL запроса для определения времени реакции сервера**	timeconnect(47), timenamelookup(48), timepretransfer(49), timestarttransfer(50), timetotal(51)

— Если ни одна из целей уровня обслуживания не была нарушена, то состояние КС равно маркеру ***normal***.

— Если (*ping_avg*) – время ответа более 5 миллисекунд, то состояние КС равно маркеру ***signal_delay.***

— Если (*server_response_timetotal*) – время ответа более 1.5 секунды, то состояние КС равно маркеру ***server_response_delay***.

— Если (*network_outdropped*) – количество пакетов более 0 шт., то состояние КС равно маркеру ***packets_dropped***.

— Если (*disk_ioreadmergespersec*) – время обработки запроса более 2 секунд, то состояние КС равно маркеру ***disk_iowriteawait***.

Допускается одновременное возникновение нескольких аномальных состояний КС, при котором наблюдается превышение нескольких порогов SLO. Целесообразно предусмотреть каскадное преобразование данных следующего вида (рис.4): «четырехмерный временной ряд метрического типа» (рис.4а)) — «пятимерный временной ряд категориального типа» (рис. 4б)) — «одномерный временной ряд категориального типа» (рис.4в)).

Распределение экспериментальных данных по числу одновременно нарушаемых показателей SLO представлены в таблице табл. 4. Распределение экспериментальных данных по каждой из комбинаций состояний КС представлены в табл. 5.

Результатом предобработки экспериментальных данных будет маркировка каждой записи в «исторических данных» одним из 12 состояний КС из табл. 5. В результате численных экспериментов были определены следующие границы изменения стартовых параметров АПМЗ:

— Стартовая точка: 4744;

— горизонт *gor*=4744;

— шаг прогнозирования *step*, $0 < step \leq 256$;

— диапазон, в котором меняется размер масок $minLength=1$, $minLength \leq maxLength \leq 5$;

— порог веса взвешенного шаблона $state\_level(т)$, $0 < state\_level(т) < 50$

— порог надежности взвешенного шаблона *reliability_level*, $0 \leq reliability\_level \leq 30$;

— порог голосования *wdifference*, $0 \leq wdifference \leq 1$;

— минимальный порог голосования для выдачи прогноза *votelevel* $0 \leq votelevel \leq 1$;

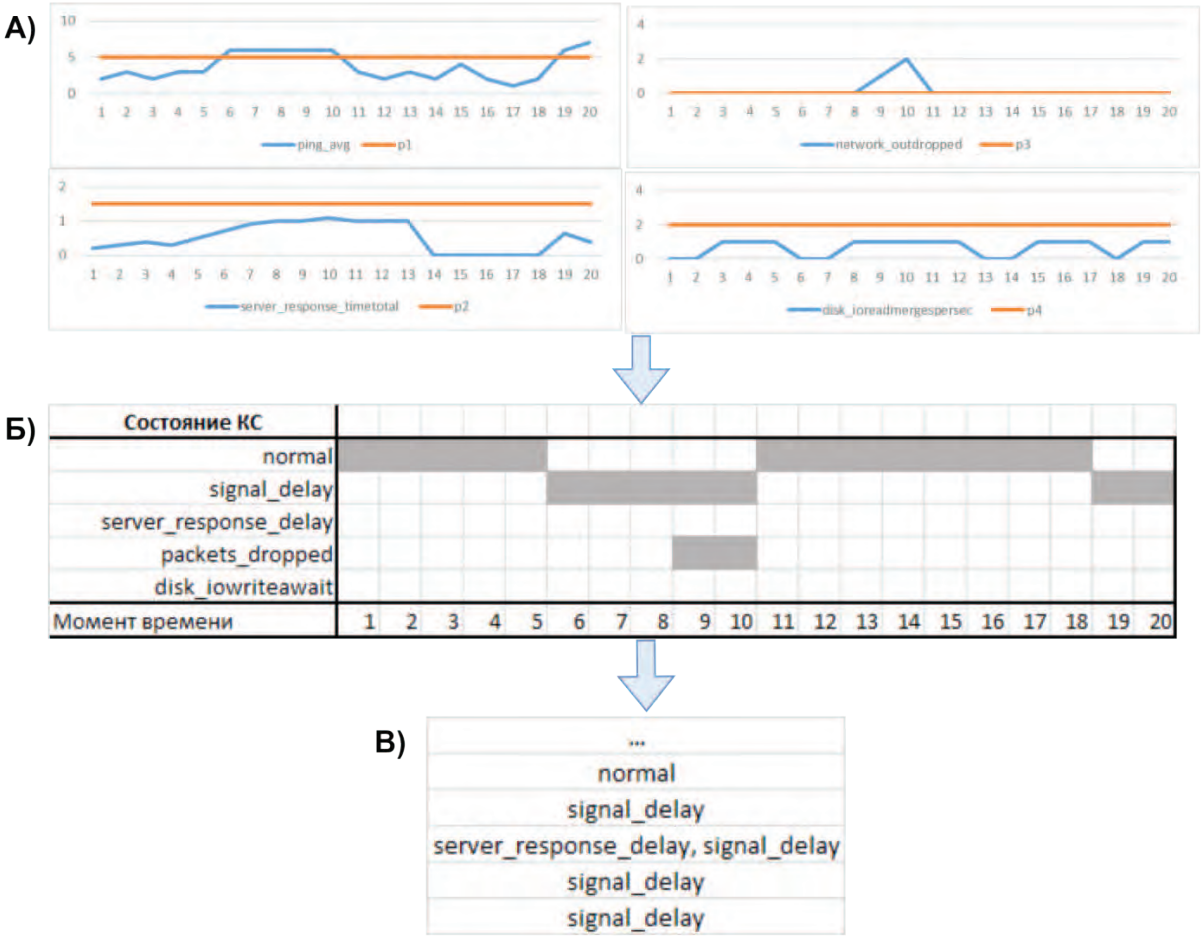

Рис. 4. Преобразование метрических экспериментальных данных в категориальные значения:
а) — «четырехмерный временной ряд метрического типа»;
б) — «пятимерный временной ряд категориального типа»;
в) — «одномерный временной ряд категориального типа».

Таблица 4

Распределение экспериментальных записей по числу одновременно нарушаемых показателей уровня обслуживания КС

Число одновременно нарушаемых показателей уровня обслуживания, Anomaly	**Количество записей в экспериментальных данных, ед.**	**Количество записей в экспериментальных данных, %**	
0	170931	71,870	
1	60447	25,416	0.28057
2	6282	2,641	
3	175	0,074	
4	0	0	

Таблица 5

Распределение экспериментальных данных по каждой из комбинаций состояний КС

№	Состояние КС	Число записей, шт.	Доля записей в выборке
1	*normal*	170931	0.7187
2	*packets_dropped*	29294	0.1232
3	*server_response_delay*	25705	0.1081
4	*server_response_delay packets_dropped*	4674	0.0197
5	*disk_iowriteawait*	4209	0.0177
6	*signal_delay*	1239	0.0052
7	*packets_dropped disk_iowriteawait*	727	0.0031
8	*signal_delay server_response_delay*	473	0.00198
9	*signal_delay packets_dropped*	234	0.0009
10	*server_response_delay disk_iowriteawait*	174	0.0007
11	*signal_delay server_response_ delay packets_dropped*	121	0.0005
12	*server_response_delay packets_ dropped disk_iowriteawait*	54	0.0002
	Итого:	237 835	1

— δ, $0.5 < \delta \leq 1$, коэффициент учета влияния удаленности шаблона от настоящего текущего времени;

— δ, $0.5 < \delta \leq 1$ – коэффициент учета удаленности шаблона внутри набора шаблонов.

Исходя из заданных параметров, анализировались первые 5 тысяч записей «исторических данных». Выборка была разделена на две части – обучающую (4477 записей, ~95%) и тестовую (256 записей, ~5%).

Количество комбинаций состояний КС в выбранном диапазоне «исторических данных» оказалось равным 8 шт., что составляет 67% от всего объема возможных комбинаций (полное количество комбинаций составляет 12 шт., см. табл. 5). Тестовая выборка не содержала уникальных комбинаций КС, отсутствующих в обучающем наборе записей и содержала записи о одновременном превышении двух и трех порогов SLO (редкое аномальное состояние КС). Комбинации состояний КС, включенных в выборку, содержала следующие уникальные состояния КС:

normal; packets_dropped; server_response_delay; server_response_delay; packets_dropped;

Наглядной трактовкой результатов прогнозирования траектории профиля функционирования КС во времени является пространственная визуализация, содержащая информацию о расстояниях между точками, ассоциированными с комбинацией метрических атрибутов и точками, ассоциированными с прогнозным состоянием КС.

Для визуализации процесса миграции профиля функционирования КС во времени необходимо отобразить траекторию движения профиля функционирования КС в пространстве атрибутов с сокращенной размерностью.

Наилучшую наглядность среди рассмотренных алгоритов (PCA (англ principal component analysis) , LDA (англ. Latent Dirichlet allocation, SVD (англ. Singular Value Decomposition), KPCA (KernelPCA)) показал алгоритм машинного обучения для визуализации t-SNE (англ. t-distributed Stochastic Neighbor Embedding). С этой целью из данных алгоритмом t-SNE была удалена временная составляющая, существенно затрудняющая преобразование данных в разделимые кластеры при сворачивании пространства атрибутов в двухмерное пространство. Преобразованный набор исторических данных и полученный прогноз, сокращенный до размерности в два измерения, с нанесенным на фоне графиком оценки плотности ядра KDE (англ.

kernel density estimate) [23] приведен на рис. 5. Точки маркированы цветом, ассоциированным с определенным состоянием КС.

На рисунке наблюдаются области повышенной плотности, расположенные в виде двухмерного тороида. В левой части изображения наблюдается кластер из точек, ассоциированный с профилем нормального функционирования КС, отмеченный синим цветом. По соседству расположены кластеры, ассоциированные с аномальными состояниями КС (server_responce_delay – оранжевый цвет, signal_delay – коричневый цвет). Между данными кластерами находятся точки, соответствующие аномалии, наблюдаемой одновременно по двум атрибутам (server_responce_delay и signal_delay – розовый цвет). На удалении от кластера с точками, ассоциированными с профилем нормального функционирования, расположены точки, ассоциированные с аномалией по атрибуту (packet_dropped, красный цвет). Зеленым цветом выделены точки, ассоциированные с аномалиями, одновременно наблюдаемыми по атрибутам server_responce_delay и packet_dropped.

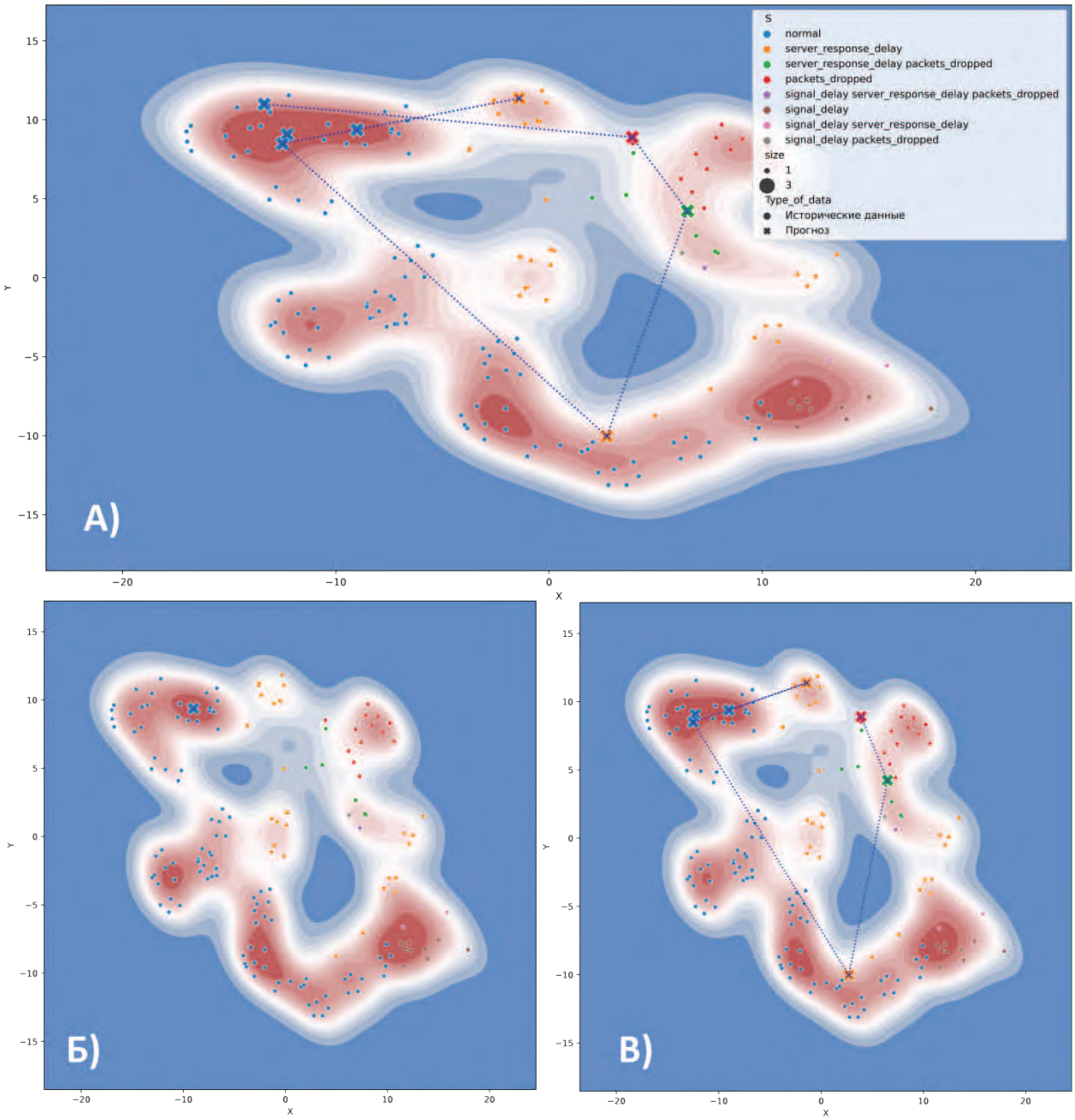

Рис. 5. Преобразованный алгоритмом t-SNE набор исторических и прогнозных данных, сокращенный до размерности в 2 измерения с нанесенным на фоне графиком оценки плотности ядра (KDE) с демонстрацией траектории изменения состояния КС

На рисунке 5а приведена траектория движения профиля функционирования КС за 256 прогнозных точек «в будущее». Рисунок 5б содержит первые 51 спрогнозированные точки профиля функционирования КС, относящиеся к профилю нормального функционирования КС. На рисунке 5в отражена эволюция движения профиля функционирования КС от нормального в аномальные состояния по двум из четырех атрибутам (server_responce_delay и packet_dropped).

Анализ диаграммы по оценке плотности ядра позволяет сделать вывод о неравномерном распределении «исторических данных» в пространстве атрибутов. Выбранный метод свертки пространства атрибутов позволил наглядно визуализировать разделимость кластеров, ассоциированных с разными состояниями КС. При этом пространство атрибутов при выбранных стартовых параметрах распределилось в виде тороида.

По траектории движения спрогнозированного профиля функционирования КС возможно оценить переход из состояния нормального функционирования КС в состояние нарушения функционирования только по одному из атрибутов, или в состояние нарушения функционирования по двум атрибутам. Эксперимент показал, что наиболее значимой информацией о профиле функционирования КС является этап перехода из состояния нормального функционирования в аномальное.

В пространственной трактовке результатов прогнозирования профиля функционирования КС целесообразно перейти от задачи кластеризации к задаче классификации состояний КС на основе ядерного преобразования пространства атрибутов (например, SVDD (Support Vector Data Description) [24]).

На рис. 6 приведены результаты построения классифицирующей гиперсферы вокруг точек, ассоциированных с профилем полностью нормального функционирования КС в трех проекциях. На трехмерную проекцию (рис. 6а) нанесены границы полученной маркирующей гиперсферы с учетом плотности распределения данных, а также информация о плотности точек внутри класса в виде поверхности. Двухмерная проекция поверхности с сохранением информации о плотности точек приведена на рис. 6б. На рисунке 6в изображена двухмерная проекция поверхности с нанесенными положительными, отрицательными классами и поддерживающими векторами. Точки, ассоциированные с нормальным состоянием КС, маркированы синими плюсами. Остальные состояния КС — маркированы фиолетовыми крестами. Точки, ассоциированные с опорными векторами (Support vectors) маркированы зелеными точками.

На рис. 7 отображены результаты классификации для классификации четырех рассмотренных выше про-

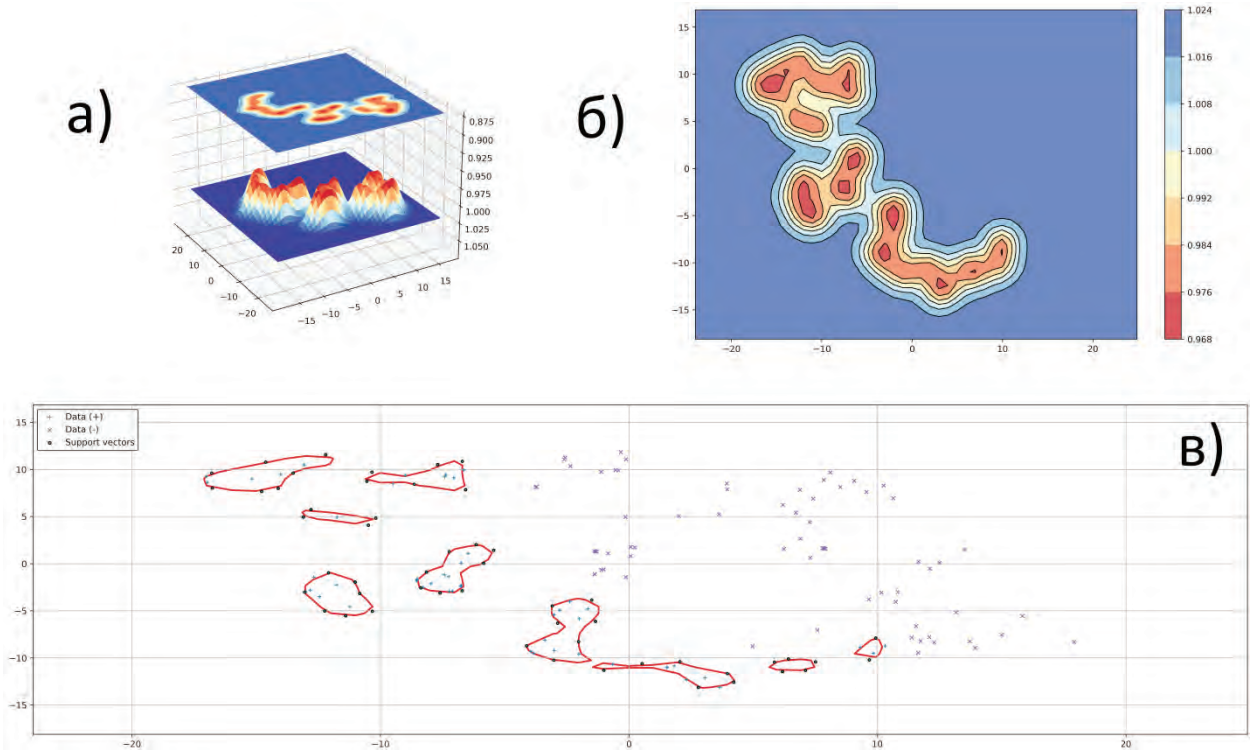

Рис. 6. Результаты построения классифицирующей гиперсферы вокруг точек, ассоциированных с профилем полностью нормального функционирования КС

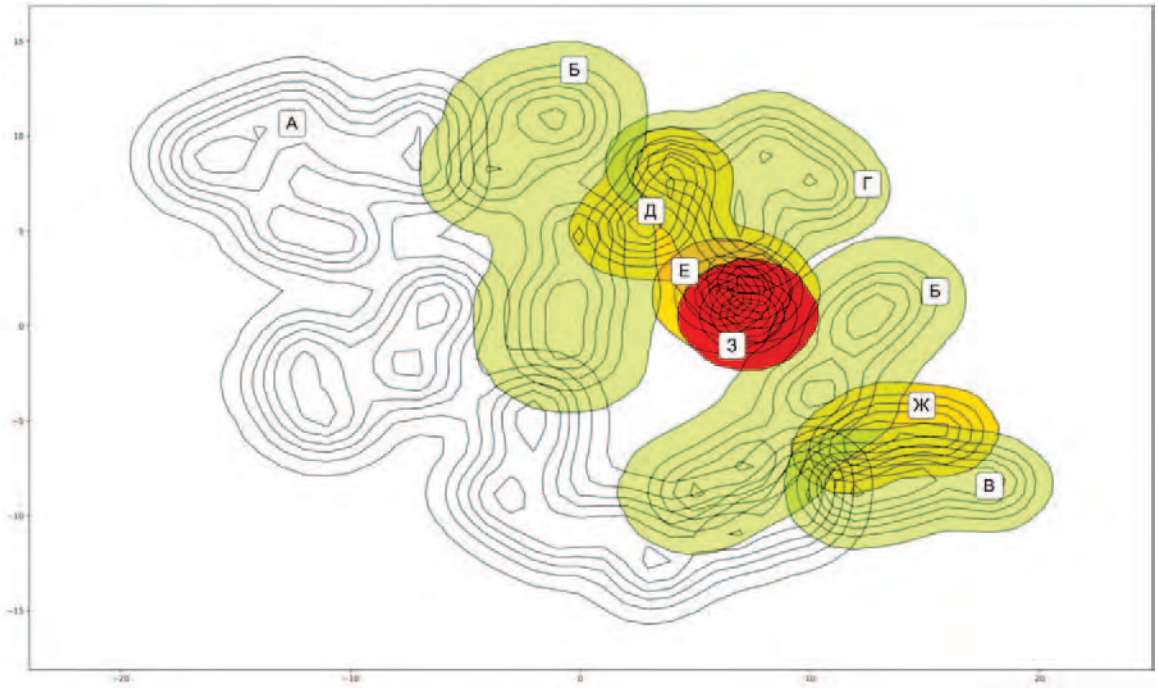

Рис. 7. Результаты классификации алгоритмом SVDD различных профилей функционирования КС

филей функционирования КС после применения алгоритма SVDD. Белым цветом маркирован профиль полностью нормального функционирования КС (normal, рис. 7а). Зеленым цветом маркировано состояние нарушения SLO по 1 атрибуту *(server_response_delay,* рис. 7б; *signal_delay,* 13в; *packets_dropped,* 7г). Желтым маркирован профиль КС, в котором наблюдается одновременное нарушение SLO по 2 атрибутам *(server_response_delay packets_dropped,* 7д; *signal_delay packets_dropped* 7е; *signal_delay server_response_delay* 7ж). Состояние одновременного нарушения SLO по 3 атрибутам *(signal_delay, server_response_delay, packets_dropped)* приведено на рис. 7з.

Полученные результаты классификации показывают, что зоны аномального функционирования КС по нескольким атрибутам расположены на пересечении кластеров наблюдения аномалии по одному атрибуту. Исследование характера изменения состояния КС (рис. 5) показывает эволюцию числа атрибутов, по которым наблюдается аномалия в соответствующий момент времени.

Заключение

Предложена методология построения алгоритма прогнозирования аномальных состояний компьютерной системы АПМЗ, характеризуемой категориальными атрибутами, заданными в виде конечного семейства многозначных закономерностей, порождаемых таблицей «исторических данных». Специфика метода состоит в том, что наиболее подходящие закономерности выбираются не заранее, а в текущем времени для каждого текущего набора значений аргументов. Использование многозначных закономерностей позволяет отказаться от априорных гипотез о модели данных, что делает метод универсальным.

На основе предложенной методологии разработан алгоритм АПМЗ, состоящий из 9 этапов. Выведены и обоснованы границы входных параметров алгоритма, которые необходимо настроить для корректной выдачи прогноза. Разработана программная реализация предложенного алгоритма прогнозирования.

Разработанный алгоритм прогнозирования (как и сам математический аппарат точечно-множественных отображений многозначных закономерностей), может быть обобщен на любую предметную область, содержащую «исторические данные» — экономика, логистика, кибернетика и т.д. Тип данных не имеет значения – прогнозироваться могут как категориальные, так и метрические значения функции.

Работоспособность алгоритма проанализирована для реальных экспериментальных данных и конкретных значений входных параметров.

Основным недостатком предложенного алгоритма является необходимость точной настройки входных параметров под каждый набор «исторических данных».

Пространственный анализ результатов прогнозирования по оценке плотности ядра позволяет сделать вывод о неравномерном распределении «исторических данных» в пространстве атрибутов.

Литература

1. Гайфулина Д.А., Котенко И.В. Применение методов глубокого обучения в задачах кибербезопасности. Часть 1 // Вопросы кибербезопасности. 2020. № 3 (37). С. 76-86. DOI: 10.21681/2311-3456-2020-03-76-86
2. Гайфулина Д.А., Котенко И.В. Применение методов глубокого обучения в задачах кибербезопасности. Часть 2 // Вопросы кибербезопасности. 2020. № 4 (38). С. 11-21. DOI: 10.21681/2311-3456-2020-04-11-21
3. Емалетдинова Л.Ю., Мухаметзянов З.И., Катасёва Д.В., Кабирова А.Н. Метод построения прогнозной нейросетевой модели временного ряда // Компьютерные исследования и моделирование. 2020. № 4. С. 737-756. DOI: 10.20537/2076-7633-2020-12-4-737-756
4. Цымблер М.Л., Краева Я.А. Параллельный алгоритм поиска лейтмотивов временного ряда для графического процессора // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. 2020. № 3. С. 17-34. DOI: 10.14529/cmse200302
5. Shatnawi M., Hefeeda M. Real-time failure prediction in online services // IEEE Conference on Computer Communications (INFOCOM). 2015. С. 1391–1399. DOI: 10.1109/INFOCOM.2015.7218516
6. Шелухин О.И., Осин А.В., Костин Д.В. Мониторинг и диагностика аномальных состояний компьютерной сети на основе изучения “исторических данных” // T-Comm: Телекоммуникации и транспорт. 2020. №4. С. 23-30. DOI: 10.36724/2072-8735-2020-14-4-23-30
7. Sheluhin O.I., Kostin D.V., Polkovnikov M.V. Forecasting of Computer Network Anomalous States Based on Sequential Pattern Analysis of “Historical Data” // Automatic Control and Computer Sciences. 2021. № 6. С. 522–533. DOI: 10.3103/S0146411621060067
8. Williams, B. A., Catherine F. B., Shmargad Y. How Algorithms Discriminate Based on Data They Lack: Challenges, Solutions, and Policy Implications. // Journal of Information Policy. 2018. №8. С. 78–115. DOI: 10.5325/jinfopoli.8.2018.0078.
9. Graber C., Meshi O., Schwing A. Deep structured prediction with nonlinear output transformations // 32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada. 2018. С. 1 – 14. DOI: 10.48550/arXiv.1811.00539
10. Zhang H., Liu J., Li K., Tan H., Wang G. Gait learning based authentication for intelligent things // IEEE Transactions on Vehicular Technology. 2020. № 4. С. 4450-4459. DOI: 10.1109/TVT.2020.2977418
11. Bodyanskiy Y., Boiko O. Online fuzzy clustering of data streams // Studies in Computational Intelligence. 2020. № 876. С. 211-241. DOI: 10.1007/978-3-030-35480-0_5
12. YanPing Z., XiaoLai Z. K-means Clustering Algorithm and Its Improvement Research // Journal of Physics: Conference Series. 2020. № 1873(1):012074 С. 1-6. DOI: 10.1088/1742-6596/1873/1/012074.
13. Yingwen Z., Songcan C. Growing neural gas with random projection method for high-dimensional data stream clustering // Soft Computing. 2020. № 24. С. 1-19. DOI:10.1007/s00500-019-04492-4.
14. Amos B., Xu L., Kolter J. Z. Input convex neural networks // Proceedings of the 34th International Conference on Machine Learning. 2017. № 70. С. 146–155.
15. M. Gygli, M. Norouzi, Angelova A. Deep value networks learn to evaluate and iteratively refine structured outputs // Proceedings of the 34-th International Conference on Machine Learning, Sydney, Australia. 2017. С. 1341–1351. DOI: 10.48550/arXiv.1703.04363.
16. Молодцов Д. А. Экстраполяция многозначных зависимостей // Нечеткие системы и мягкие вычисления. 2017. № 1. С. 45-63.
17. Молодцов Д. А., Осин А. В. Новый метод применения многозначных закономерностей // Нечеткие системы и мягкие вычисления. 2020. № 2. С. 83-95. DOI 10.26456/fssc72
18. Rastegari Y., Shams F. Optimal Decomposition of Service Level Objectives into Policy Assertions // The Scientific World Journal. 2015. № 3. С. 1-9. DOI:10.1155/2015/465074
19. Молодцов Д. А. Сравнение и продолжение многозначных зависимостей // Нечеткие системы и мягкие вычисления. 2016. №2. С. 115–145
20. Шелухин О.И., Раковский Д.И. Выбор метрических атрибутов редких аномальных событий компьютерной системы методами интеллектуального анализа данных // T-Comm: Телекоммуникации и транспорт. 2021. № 6. С. 40-47 DOI: 10.36724/2072-8735-2021-15-6-40-47
21. Шелухин О.И., Осин А.В., Костин Д.В. Диагностика «здоровья» компьютерной сети на основе секвенциального анализа последовательностных паттернов // T-Comm: Телекоммуникации и транспорт. 2020. №2. С. 9-16. doi:10.36724/2072-8735-2020-14-2-9-16
22. Шелухин О.И., Раковский Д.И. Бинарная классификация многоатрибутных размеченных аномальных событий компьютерных систем с помощью алгоритма SVDD // Наукоемкие технологии в космических исследованиях Земли. 2021. № 2. С. 74-84. DOI: 10.36724/2409-5419-2021-13-2-74-84
23. Bose A., Bhattacharjee M. Kernel density estimates in a non-standard situation // Journal of Statistical Theory and Practice. 2021. № 1. С. 22. DOI: 10.1007/s42519-020-00161-0
24. Lv Y., Zhang J., Qin W., Yang J. Adjustment mode decision based on support vector data description and evidence theory for assembly lines // Industrial Management & Data Systems. 2018. №. 8. С. 1711-1726. DOI: 10.1108/IMDS-01-2017-0014

PREDICTION OF THE PROFILE FUNCTIONING OF A COMPUTER SYSTEM BASED ON MULTIVALUED PATTERNS

Sheluhin O.I.[3], Rakovskiy D.I.[4]

Purpose of work *is to create a new algorithm for predicting anomalous states of computer systems (CS) using the mathematical apparatus of multivalued dependencies (Multivalued Dependencies Prognosus Algorithm, MDPA), which are categorical concepts.*

The research method *is the analysis of historical data using the mathematical apparatus of multivalued dependencies.*

Objects of study *are theoretical and practical issues of solving and visualizing information security problems.*

Results of the study. *A methodology and algorithm for predicting the state of CS have been developed. The boundaries of the input parameters of the algorithm are derived and justified. The boundaries of the input parameters need to be pre-configured for the correct generation of the prognosis.*

A software implementation of the proposed prediction algorithm has been developed. The efficiency of the algorithm has been tested on real experimental data. A spatial analysis of the prediction results was carried out.

The main disadvantage of the proposed algorithm is the need to fine-tune the input parameters for each set of "historical data".

Scientific significance. *The scope of application of multivalued dependencies has been expanded; a new algorithm for predicting anomalous states of CS, which are categorical concepts, has been proposed. The developed prediction algorithm can be generalized to any subject area containing historical data of any type.*

Keywords: *historical data, time series analysis, forecasting model, time series forecasting, computer system, anomaly forecast, system state.*

References

1. Gayfulina D.A., Kotenko I.V. Primenenie metodov glubokogo obucheniya v zadachakh kiberbezopasnosti. Chast' 1 // Voprosy kiberbezopasnosti. 2020. No 3 (37). Pp. 76-86. DOI: 10.21681/2311-3456-2020-03-76-86
2. Gayfulina D.A., Kotenko I.V. Primenenie metodov glubokogo obucheniya v zadachakh kiberbezopasnosti. Chast' 2 // Voprosy kiberbezopasnosti. 2020. No 4 (38). Pp. 11-21. DOI: 10.21681/2311-3456-2020-04-11-21
3. Emaletdinova L.Yu., Mukhametzyanov Z.I., Kataseva D.V., Kabirova A.N. Metod postroeniya prognoznoy neyrosetevoy modeli vremennogo ryada // Komp'yuternye issledovaniya i modelirovanie. 2020. No 4. Pp. 737-756. DOI: 10.20537/2076-7633-2020-12-4-737-756
4. Tsymbler M.L., Kraeva Ya.A. Parallel'nyy algoritm poiska leytmotivov vremennogo ryada dlya graficheskogo protsessora // Vestnik Yuzhno-Ural'skogo gosudarstvennogo universiteta. Seriya: Vychislitel'naya matematika i informatika. 2020. № 3. Pp. 17-34. DOI: 10.14529/cmse200302
5. Shatnawi M., Hefeeda M. Real-time failure prediction in online services // IEEE Conference on Computer Communications (INFOCOM). 2015. Pp. 1391–1399. DOI: 10.1109/INFOCOM.2015.7218516
6. Sheluhin, O.I., Osin, A.V., Kostin, D.V. Monitoring and diagnostics of anomalous states in a computer network based on the study of "historical data", T-Comm. 2020. No. 4. Pp. 23–30. DOI: 10.36724/2072-8735-2020-14-4-23-30
7. Sheluhin O. I., Kostin D. V., Polkovnikov M. V. Forecasting of Computer Network Anomalous States Based on Sequential Pattern Analysis of "Historical Data" // Automatic Control and Computer Sciences. 2021. No 6. Pp. 522–533. DOI: 10.3103/S0146411621060067
8. Williams, B. A., Catherine F. B., Shmargad Y. How Algorithms Discriminate Based on Data They Lack: Challenges, Solutions, and Policy Implications. // Journal of Information Policy. 2018. No8. Pp. 78–115. DOI: 10.5325/jinfopoli.8.2018.0078.
9. Graber C., Meshi O., Schwing A. Deep structured prediction with nonlinear output transformations // 32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada. 2018. Pp. 1 – 14. DOI: 10.48550/arXiv.1811.00539
10. Zhang H., Liu J., Li K., Tan H., Wang G. Gait learning based authentication for intelligent things // IEEE Transactions on Vehicular Technology. 2020. No 4. Pp. 4450-4459. DOI: 10.1109/TVT.2020.2977418
11. Bodyanskiy Y., Boiko O. Online fuzzy clustering of data streams // Studies in Computational Intelligence. 2020. No 876. Pp. 211-241. DOI: 10.1007/978-3-030-35480-0_5

3 Oleg I. Sheluhin, Dr.Sc. (in Tech.), Professor, Head of Department Information Security of the Moscow Technical University of Communications and Informatics, E-mail: sheluhin@mail.ru;

4 Dmitry I. Rakonsky, Moscow Technical University of Communication andInformatics, E-mail: Prophet_alpha@mail.ru.

12. YanPing Z., XiaoLai Z. K-means Clustering Algorithm and Its Improvement Research // Journal of Physics: Conference Series. 2020. No 1873(1):012074. Pp. 1-6. DOI: 10.1088/1742-6596/1873/1/012074.
13. Yingwen Z., Songcan C. Growing neural gas with random projection method for high-dimensional data stream clustering // Soft Computing. 2020. No 24. Pp. 1-19. DOI:10.1007/s00500-019-04492-4.
14. Amos B., Xu L., J. Z. Kolter Input convex neural networks // Proceedings of the 34th International Conference on Machine Learning. 2017. No 70. Pp. 146–155.
15. M. Gygli, M. Norouzi, Angelova A. Deep value networks learn to evaluate and iteratively refine structured outputs // Proceedings of the 34-th International Conference on Machine Learning, Sydney, Australia. 2017. Pp. 1341–1351. DOI: 10.48550/arXiv.1703.04363.
16. Molodtsov, D. A. Ekstrapolyatsiya mnogoznachnykh zavisimostey // Nechetkie sistemy i myagkie vychisleniya. 2017. No 1. Pp. 45-63.
17. Molodtsov D. A., Osin A. V. Novyy metod primeneniya mnogoznachnykh zakonomernostey // Nechetkie sistemy i myagkie vychisleniya. 2020. No 2. Pp. 83-95. DOI 10.26456/fssc72
18. Rastegari Y, Shams F. Optimal Decomposition of Service Level Objectives into Policy Assertions // The Scientific World Journal. 2015. No 3. Pp. 1-9. DOI:10.1155/2015/465074
19. Molodtsov D. A. Sravnenie i prodolzhenie mnogoznachnykh zavisimostey // Nechetkie sistemy i myagkie vychisleniya. 2016. No 2. Pp. 115–145
20. Shelukhin O.I., Rakovskiy D.I. Vybor metricheskikh atributov redkikh anomal'nykh sobytiy komp'yuternoy sistemy metodami intellektual'nogo analiza dannykh // T-Comm: Telekommunikatsii i transport. 2021. No 6. Pp. 40-47 DOI: 10.36724/2072-8735-2021-15-6-40-47
21. Shelukhin O.I., Osin A.V., Kostin D.V. Diagnostika "zdorov'ya" komp'yuternoy seti na osnove sekventsial'nogo analiza posledovatel'nostnykh patternov // T-Comm: Telekommunikatsii i transport. 2020. No 2. Pp. 9-16. doi:10.36724/2072-8735-2020-14-2-9-16
22. Shelukhin O.I., Rakovskiy D.I. Binarnaya klassifikatsiya mnogoatributnykh razmechennykh anomal'nykh sobytiy komp'yuternykh sistem s pomoshch'yu algoritma SVDD // Naukoemkie tekhnologii v kosmicheskikh issledovaniyakh Zemli. 2021. No 2. Pp. 74-84. DOI: 10.36724/2409-5419-2021-13-2-74-84
23. Bose A., Bhattacharjee M. Kernel density estimates in a non-standard situation // Journal of Statistical Theory and Practice. 2021. No 1. Pp. 22. DOI: 10.1007/s42519-020-00161-0
24. Lv Y., Zhang J., Qin W., Yang J. Adjustment mode decision based on support vector data description and evidence theory for assembly lines // Industrial Management & Data Systems. 2018. No. 8. Pp. 1711-1726. DOI: 10.1108/IMDS-01-2017-0014