ДИАГНОСТИКА "ЗДОРОВЬЯ" КОМПЬЮТЕРНОЙ СЕТИ НА ОСНОВЕ СЕКВЕНЦИАЛЬНОГО АНАЛИЗА ПОСЛЕДОВАТЕЛЬНОСТНЫХ ПАТТЕРНОВ

Шелухин Олег Иванович,

Московский технический университет связи
и информатики, Москва, Россия, sheluhin@mail.ru

Осин Андрей Владимирович,

Московский технический университет связи
и информатики, Москва, Россия, osin_a_v@mail.ru

Костин Денис Владимирович,

Московский технический университет связи
и информатики, Москва, Россия, d.v.kostin@mail.ru

DOI: 10.36724/2072-8735-2020-14-2-9-16

Ключевые слова: аномальные состояния, компьютерной сети, прогнозирование, машинное обучение, интеллектуального анализа данных, мониторинг системных показателей, кластеризация, секвенциальный анализ, паттерн.

Для решения задач диагностики "здоровья" компьютерной сети (КС) предложено, использовать методы и алгоритмы статистической обработки данных, машинного обучения и интеллектуального анализа "прошлых" данных полученных путем изучения истории поведения сети "в прошлом" и формализации этой информации в виде некоторых шаблонов (паттернов). В качестве метода прогнозирования аномальных состояний КС предложено использовать метод выбора из полученных закономерностей наиболее приемлемой с точки зрения "здоровья сети" путем использования аппарата секвенциального анализа сформированных последовательностных паттернов. Рассмотрены этапы процесса прогнозирования на основе последовательностных паттернов и структура реализованного алгоритма вычисления прогноза текущего состояния КС. В качестве целевой функции оптимизации гиперпараметров алгоритма прогнозирования предложено использовать функцию Парето. В качестве лучшего набора параметров выбирается тот, при котором достигается максимальное значение введенной целевой функции. В качестве системных показателей, характеризующих "здоровье компьютерной сети" используется уровень обслуживания (Service Level Objectives, SLO), и соглашение об уровне предоставляемого сервиса (Service Level Agreement, SLA). На конкретном примере рассмотрена визуализация процесса кластеризации состояний КС с использованием алгоритма k-means и алгоритма понижения размерности – TSNE. Предложено оценивать "здоровье сети" по близости текущей оценки прогноза состояния КС к сформированной в результате кластеризации области аномальных состояний в виде расстояния до ближайших центров кластеров, на порядковой шкале от 1 до 5. Предложено характеризовать "состояние здоровья" КС с помощью "Зеленого", "Желтого" и "Красного" уровней аномальности.

Информация об авторах:

Шелухин Олег Иванович, д.т.н., МТУСИ, Москва, Россия

Осин Андрей Владимирович, к.т.н., МТУСИ, Москва, Россия

Костин Денис Владимирович, аспирант МТУСИ, Москва, Россия

Для цитирования:

Шелухин О.И., Осин А.В., Костин Д.В. Диагностика "здоровья" компьютерной сети на основе секвенциального анализа последовательностных паттернов // Т-Comm: Телекоммуникации и транспорт. 2020. Том 14. №2. С. 9-16.

For citation:

Sheluhin O.I., Osin A.V., Kostin D.V. (2020) Health monitoring of a computer network based on sequential analysis of serial pattern. T-Comm, vol. 14, no.2, pp. 9-16. (in Russian)

Постановка задачи

Обнаружение аномалий играет важную роль в управлении современными крупномасштабными распределенными компьютерными системами [1,2,3]. Системные журналы, в которых записывается информация о времени работы системы, широко используются для обнаружения аномалий [4,5]. Тем не менее, возрастающий масштаб и сложность современных систем приводят к объему логов, что делает невозможным ручную проверку.

Обнаружение аномального поведения системы, играет важную роль в управлении инцидентами крупномасштабных систем. Своевременное обнаружение аномалий позволяет разработчикам системы (или операторам) быстро выявлять проблемы и немедленно их устранять, тем самым сокращая время простоя системы.

Одной из важных задач анализа данных является выделение закономерностей [6,7]. Для последовательных данных эту задачу решает в частности такая область интеллектуального анализа данных (data mining) как секвенциальный анализ или анализ последовательностных паттернов (sequential pattern mining) [8,9,10], позволяющих выявлять часто встречающиеся участки в строках и в последовательностях наборов элементов.

Добыча последовательностных паттернов является частью интеллектуального анализа данных, связанная с поиском статистически значимых шаблонов (паттернов) в данных, где значения можно представить в виде последовательности [11]. Последовательный анализ паттернов состоит из обнаружении интересных подпоследовательностей в наборе последовательностей, где значимость той или иной подпоследовательности может быть измерена с точки зрения различных критериев, таких как частота появления, длина и др. Последовательный анализ шаблонов имеет множество реальных применений, поскольку данные кодируются в виде последовательностей во многих областях, таких как биоинформатика, электронное обучение, анализ корзины, анализ текста и анализ кликов на веб-страницах [12,13,14].

В [15] рассмотрена общая модель последовательного интеллектуального анализа с прогрессивной базой данных. Данные в базе данных могут быть статическими, вставленными или удаленными. В работе предлагается алгоритм прогрессивного установления последовательных паттернов (Progressive mining of Sequential pAttems – Pisa), для постепенного обнаружения последовательных паттернов в определенный период интереса (period of interest – POI). POI – это скользящее окно, непрерывно изменяющееся с течением времени. Алгоритм Pisa использует последовательное дерево для эффективного обслуживания последних последовательностей данных, обнаружения полного набора актуальных последовательных шаблонов и, соответственно, удаляющее устаревшие данные и шаблоны.

В [16] рассмотрен последовательный анализ шаблонов на прогрессивных базах данных, в котором на «интересующем периоде» постепенно обнаруживаются последовательные закономерности. «Интересующий период» – это скользящее окно, непрерывно продвигающееся с течением времени. При изменении фокуса скользящего окна в интересующий набор данных добавляются новые элементы, а устаревшие элементы удаляются. В статье предложен новый подход, с использованием прогрессивного дерева майнинга.

В [17] предложен алгоритм, использующий для каждой возможной комбинации элементов или набора элементов понятие «процент участия» вместо частоты появления. Концепция процента участия рассчитывается на основе минимального порога поддержки для каждого установленного пункта. В э статье представлен алгоритм MS DirApp, который расшифровывается как Multiple Support Direct Appending, обнаруживающий последовательные шаблоны, рассматривая различные множественные минимальные пороговые значения поддержки для каждой возможной комбинации элементов или наборов элементов.

Основная идея прогнозирования «здоровья» компьютерной сети (КС) рассмотренная в статье заключается в изучении исторических данных о поведении сети «в прошлом» и формализации этой информации в виде некоторых шаблонов (паттернов).

Особенности задачи прогнозирования здоровья сети с использованием последовательностных паттернов

На основании полученной информации о типовых ситуациях в поведении КС «в прошлом» можно осуществлять прогнозирование появления того или иного паттерна в ближайшем будущем. Наиболее распространенный способ описание закономерностей – это использование однозначной функции

$$f: X \rightarrow Y, \quad (1)$$

значениями которой являются значения описываемой переменной x .

При практической проверке такой закономерности могут возникнуть ситуации, когда при одном и том же значении аргумента появляются результаты опытов с различными значениями функции, так что применение однозначных закономерностей основано на принятии ряда дополнительных гипотез и ограничений. Альтернативой использования однозначной функции подобного соответствия вида (1) является использование многозначных отображений, для которых любой набор результатов опытов, представленных в виде таблицы

$$D_n = \{(x_1, y_1), \dots, (x_n, y_n) \mid (x_i, y_i) \in X \times Y\}, \quad (2)$$

можно рассматривать, как график точечно множественного отображения.

Здесь $X \times Y$ означает декартово произведение двух множеств X и Y , элементами которого являются все возможные упорядоченные пары «входных» – x_i и «выходных» – y_i элементов исходных множеств.

Под исходными входными данными понимается вся доступная информация о системе к текущему моменту времени. Из этой информации должна быть сформирована таблица данных, в которой один столбец будет играть роль функции, или предсказываемого параметра, а остальные столбцы или поля, как мы их будем называть, роль аргументов. Множество значений поля функции обозначим F .

Подобный выбор набора аргументов и множества «входных» данных полученных в результате опытов будет означать построение многозначной закономерности. В этом случае конечный результат прогнозирования состояния сети будет состоять не в построении закономерности вида (1), а в выборе из полученных закономерностей наиболее приемлемой с точки зрения «здоровья сети». Реализация этой идеи

приводит к следующей последовательности этапов построения прогноза.

1. Формирование таблицы исходных данных.
2. Получение новых значений исходной информации и перевод их в таблицу исходных данных.
3. Выбор наилучших многозначных закономерностей.
4. Построение результирующего значения прогноза.

Методология сравнения многозначных закономерностей

Будем предполагать, что исходным материалом для построения многозначной закономерности служит набор результатов опытов (2), где $X \subseteq X_1 \times \dots \times X_m$.

Искомая многозначная закономерность определяется указанием непустого подмножества аргументов $I \subseteq \{1, \dots, m\}$ и представляет собой мультимножество $\{X \times Y, I, \mu\}$, где

$$\mu: X(I) \times Y \rightarrow N, \quad (3)$$

где $X(I) = \prod_{i \in I} X_i$, N – множество натуральных чисел.

Значение $\mu(x, y)$ трактуется, как **надежность точки (x, y) закономерности $\{X \times Y, I, \mu\}$** и показывает сколько раз пара (x, y) ,

$$x = \prod_{i \in X(I)} x_i \in X(I), y \in Y \text{ встречалась в наборе данных } D_n.$$

Надежность всей закономерности при значении аргумента $x \in X(I)$ считается равной сумме $\sum_{y \in Y} \mu(x, y)$ и принимается равной величине

$$Saf[X \times Y, I, \mu] = \min_{x \in X(I)} \sum_{y \in Y} \mu(x, y). \quad (4)$$

Точностью закономерности $\{X \times Y, I, \mu\}$ в точке $x \in X(I)$ будем называть мощность множества $\{y \in Y | \mu(x, y) > 0\}$, которую будем обозначать символом $card\{y \in Y | \mu(x, y) > 0\}$.

Точность всей закономерности $\{X \times Y, I, \mu\}$ считается равной

$$Ac(X \times Y, I, \mu) = \max_{x \in X(I)} card\{y \in Y | \mu(x, y) > 0\}. \quad (5)$$

При построении методологии сравнения необходимо предварительно отобрать закономерности, достигающие заданного уровня надежности, а потом оставшиеся закономерности сравнивать по точности. Однако, при таком способе сравнения закономерности с большим количеством аргументов будут иметь, вообще говоря, меньшую надежность, чем закономерности с небольшим количеством аргументов.

Если закономерность имеет недостаточный уровень надежности, то для решения проблемы в [18] предложено применять к ней процедуру замены множества значений аргументов на сеточное множество с меньшим количеством значений (квантованию). Подобная процедура квантования приводит к увеличению надежности закономерности, однако может понизить ее точность.

Для того чтобы такая процедура проводилась плавно, с постепенным изменением надежности, нужно задать семейство таких сеток на множестве значений аргументов.

После достижения заданного уровня надежности закономерность сравнивается с другими по точности, но не исходной точности, а точности, соответствующей достигнутому уровню надежности. Если достигнуть нужного уровня надежности не удастся, то закономерность признается неудовлетворительной и в сравнении больше не участвует.

Найденные закономерности представляются в виде графиков, заданных конечным набором точек. Для дальнейшего использования, например, в прогнозировании, такое представление нужно дополнять экстраполяцией.

Выбор наилучших многозначных закономерностей

Выбор закономерности – это выбор непустого подмножества в множестве полей аргументов. Если таких полей N , то количество непустых подмножеств равно $2^N - 1$. При небольших значениях N можно перебирать все подмножества, при больших значениях перебор нужно ограничить. Поэтому для работы алгоритма нужно указать возможные значения для подмножества множества полей аргументов. Для краткости эти подмножества будут называться масками закономерности, а множество возможных масок обозначим $Mask$.

Для каждой маски $M \in Mask$ вычисляется вектор последних значений аргументов $Ш(T, M) \in X(M)$, где T – это последний момент времени обновления информации в данных.

Вектор $Ш(T, M)$ будем называть шаблоном (паттерном).

Для каждой допустимой маски формируется статистический список $Stat(T, M)$, который рассчитывается следующим образом:

Для каждого допустимого значения времени t проверяется условие $Ш(t, M) = Ш(T, M)$. Если условие выполнено, то в список результатов опыта (2) добавляется пара $(t, f(t))$, где $f(t)$ – значение поля функции.

На следующем этапе вычисляется надежность паттерна $Ш(T, M)$, которая равна следующей величине

$$S(T, M) = \sum_{(t, f(t)) \in Stat(T, M)} \gamma^{T-t}. \quad (6)$$

Здесь $\gamma, 0 < \gamma \leq 1$ – это коэффициент учитывающий «старение данных» и который дает возможность учитывать совпадения паттернов в зависимости от удаленности от последнего момента времени.

Коэффициент γ является параметром метода.

В случае когда параметр $\gamma = 1$ соотношение (6) принимает вид

$$S(T, M) = \sum_{(t, f(t)) \in Stat(T, M)} 1. \quad (6a)$$

Задается еще один параметр метода – уровень надежности S , который оценивается соотношением (4).

Если $S(T, M) \geq S$, то данный шаблон участвует в прогнозировании. Если неравенство не выполнено, выполняется последовательное синхронное огрубление всех полей аргументов, входящих в маску, до тех пор, пока ограничение по надёжности не будет выполнено. Огрубление аргументов выполняется с помощью операции квантования. Количество уровней и шаг квантования определяются свойствами огрубляемых процессов.

Если в результате процесса округления ограничение по надёжности так и не выполнено, то данный шаблон не участвует в построении прогноза.

При описанном способе отбора масок используется надёжность только одного значения закономерности, а именно того значения, которое реализовалось в текущий момент времени и может использоваться для прогнозирования.

Обозначим множество масок, оставшееся после такого отбора, символом $SMask(T)$.

Пример наложения маски длиной $M_M = 5$ на последовательность квантованных величин $\{a_i\}$ и текущего шаблона $Ш(t, M)$, иллюстрируется на рис. 1.

Рис. 1. Процесс наложения маски на одномерный временной ряд квантованных признаков

Результатом наложения шаблона на матрицу квантования является подсчёт количества возможных вариантов сочетания текущих номеров уровней квантования в каждом из шаблонов при перемещении матрицы шаблонов назад «в прошлое».

Построение результирующего значения прогноза

После окончания отбора паттернов формируется некоторое множество паттернов со статистикой $\{Ш(T, M), Stat(T, M), M \in SMask(T)\}$.

Для каждого паттерна считается статистика появления значений поля функции $u \in F$ с учетом коэффициента $\delta, 0 < \delta \leq 1$, характеризующего «старение» данных, по формуле

$$Q(T, M, u) = \sum_{(t, f(t)) \in Stat(T, M)} \delta^{T-t}. \quad (8)$$

Функция $Q(T, M, u)$ – это аналог функции кратности при описании многозначной закономерности, но с учетом удаленности анализируемого фрагмента по времени «в прошлое» от текущего момента времени.

На следующем этапе производится нормировка

$$q(T, M, u) = \frac{Q(T, M, u)}{\sum_{u \in F} Q(T, M, u)}. \quad (9)$$

Полученные статистики $Q(T, M, u)$ обрабатываются с целью выделения наиболее «сильных» значений. С этой целью вводится параметр «силы» – $Strong$. Для этого при анализе оставляются только те значения $Q(T, M, u)$, которые обеспечивают максимум соотношения (10) с точностью до параметра $\beta, 0 \leq \beta \leq 1$.

Вычисление производится по формуле

$$Strong(T, M) = \{u \in F \mid q(T, M, u) \geq \beta \max_{v \in F} q(T, M, v)\}. \quad (10)$$

Остальные статистики обнуляются.

Формирование значения окончательного прогноза

Вышеизложенное показывает, что исходная информация для выработки окончательного прогноза имеет следующий вид:

- Надёжность паттернов (масок) $S(T, M), M \in SMask(T)$.
- Статистики $Q(T, M, u), u \in Strong(T, M), M \in SMask(T)$.

Результирующую статистику можно сформировать по формуле:

$$Result(T, u) = \sum q(T, M, u) \chi[Strong(T, M)](u), \quad (11)$$

Здесь символом $\chi[A](u)$ обозначена характеристическая функция множества $A \subseteq F$,

$$\chi[A](u) = \begin{cases} 1, u \in A \\ 0, u \in \overline{A} \end{cases}.$$

В зависимости от типа значения прогноза можно сформировать его значение. Например прогноз можно положить равным функции $Result(T, u)$ или предварительно нормировать эту функцию.

Алгоритм прогнозирования

Этапы промежуточных вычислений на основе последовательных паттернов представлены на рис. 2.

Рис. 2. Этапы процесса прогнозирования на основе последовательных паттернов

Алгоритм вычисления прогноза имеет вид:

Шаг 1. Задаётся вариант маски M_i .

Шаг 2. Для каждой маски задаётся вариант шаблона $Ш_j, j = 1 \dots K!(2^M M - 1)$.

Шаг 2. Для выбранной маски каждого шаблона $Ш_i$ осуществляется подсчёт числа событий, при которых после шаблона $Ш_i$ следует уровень квантования

$$a_i : N\{Ш_j, t \mid t+1, a_i\}.$$

Шаг 4. Для каждого i -го шаблона находится:

$$\text{Max} N\{Ш_j, t \mid t+1, a_i\} i = \overline{1, K}.$$

Шаг 5. Находится наиболее вероятное значения прогнозируемого уровня квантования

$$\text{Max} N\{Ш_j, t \mid t+1, a_i\} i = \overline{1, K}, j = 1 \dots K!(2^M M - 1).$$

Шаг 6. Анализируются все варианты масок и принимается решение о наиболее вероятном прогнозе значения уровня квантования на шаге $t+1$

$$a_i + 1 = a_i \max N\{Ш_j, t | t+1, a_i\}, i = \overline{1, K}, j = 1 \dots K!(2^M M - 1).$$

Для характеристики числа совпадений текущего шаблона $Ш_{ij}$ с данными вектора квантованных величин $A_j^{кв}$ при смещении маски M_i на 1 шаг (одну ячейку) “в прошлое” вводится понятие уровня надёжности шаблона. Надёжность текущего шаблона $Ш_i$ в текущей момент времени вычисляется по следующему правилу $S_i(Ш_i) = card\{t-l, f\}$, где $card\{t-l, f\}$ – общее количество (мощность) элементов конечного множества пар сочетаний значений текущих шаблонов и исторических прогнозов (ИП); l_j – характеризует шаг сдвига маски “в прошлое”, при котором значение текущего находится шаблон $Ш_j$; f – историческое значение вектора квантования $F_j^{кв}$ в момент времени $\{t-l\}$.

Если $S_i(Ш_{ij}) \geq S_{пор}$, то данный шаблон участвует в прогнозировании.

Выбор пороговой величины $S_{пор}$ является важным этапом и состоит из нескольких шагов:

- 1) осуществляется фильтрация вычисленных значений надёжности S_i .
- 2) полученные пары максимальных значений отбраковываются путём сравнения с допустимым приращением $U_{\max 1i} - U_{\max 2i} \leq U_s$.
- 3) путём голосования определяется искомый порог $S_{пор}$.

По каждому шаблону вычисляется «количество» появления каждого исхода и вычисляется итоговый прогноз одним из следующих вариантов:

- сумма обработанных статистик (порог β) с последующей нормировкой;
- максимальные из суммы статистик по шаблону (порог α);
- взвешенная сумма статистик с весами равными надёжности статистики;
- взвешенная сумма статистик с выбором максимальных и последующей нормировкой.

В качестве целевой функции оптимизации гиперпараметров алгоритма прогнозирования предложено использовать функцию Парето:

$$F = \max \left\{ \alpha \sum_{i=1}^n \beta_i G_i + \min_{i \in \{1 \dots n\}} (\gamma_i H_i) \right\}, \quad (12)$$

где α – настраиваемый коэффициент, определяющий соотношений числа прогнозов и их точности. В общем случае данный коэффициент обратно пропорционален числу уровней квантования прогнозируемой функции; β_i – коэффициент значимости числа прогнозов, формируемых по уровню квантования n ; G_i – относительное число прогнозов к общему числу значений функции для n -го уровня квантования; γ_i – коэффициент значимости точности прогнозов, выдаваемых по уровню квантования n ; H_i – точность прогнозирования функции для уровня квантования n .

В качестве наилучшего набора параметров выбирается тот, который даёт наилучшее значение целевой функции (12). После этого оптимизация считается выполненной и алгоритм с найденным набором параметров может быть запущен.

Этап кластеризации

Состояние КС предложено оценивать по близости текущей оценки прогноза состояния сети к сформированной в результате кластеризации области аномальных состояний.

При помощи кластерного анализа можно разделить большой набор данных полученных в результате прогноза на кластеры, т. е. группы похожих объектов, что позволяет значительно сократить объем данных и облегчить последующую работу с ними. При решении задачи кластеризации, в отличие от классификации, одним из первых этапов является определение зависимой переменной. Количество кластеров может быть не задано изначально и меняться в зависимости от необходимости. В анализируемом случае множество атрибутов задается, как и в задаче классификации [16] в виде вектора $I = \{i_1, i_2, \dots, i_n\}$, где i_j исследуемый атрибут, характеризующийся набором параметров (уровней квантования) $I_j = \{x_1, x_2, \dots, x_m\}$. Каждый параметр x_i принимает значение из некоторого множества $xh = \{v_h^1, v_h^2, \dots\}$. Для решения задачи кластеризации необходимо построить множество $C = \{c_1, c_2, \dots, c_g\}$, где c_k – кластер, содержащий похожие друг на друга атрибуты:

$$c_k = \{i_j, i_p | i_j \in I, i_p \in I \text{ и } d(i_j, i_p) < \sigma\}, \quad (13)$$

где $d(i_j, i_p)$ – мера близости между атрибутами; σ – заданное расстояние между атрибутами для включения их в один кластер.

В качестве простого метода кластеризации рассмотрим метод k средних [19], в котором количество кластеров задано заранее. Необходимо разделить данные на кластеры таким образом, чтобы минимизировать полную сумму квадратов расстояния от данной точки до центра кластера, к которому она принадлежит. С этой целью часто используют итеративный алгоритм:

1. Выбор k произвольных исходных центров $C = \{c_i\}$, (14)

$$\text{Где } c_i = \frac{\sum_{j=1}^d u_{ij} m_j}{\sum_{j=1}^d u_{ij}}, \text{ а } M = \{m_j\} - \text{множество данных;}$$

2. Разбиение всех объектов на k групп, наиболее близких к одному из центров $U = \{u_{ij}\}$,

$$\text{где } u_{ij} = \begin{cases} 1 & \text{при } d(m_j, c_i) = \min_{1 \leq k \leq c} d(m_j, c_k) \\ 0 & \text{иначе} \end{cases}$$

3. Вычисление новых центров кластеров, пока центры не перестанут меняться, то есть пока решение не будет удовлетворять целевой функции

$$J(M, U, C) = \sum_{i=1}^c \sum_{j=1}^d u_{ij} d_A^2(m_j, c_i) \quad J(M, U, C) = \sum_{i=1}^c \dots \quad (15)$$

В качестве примера рассмотрим результаты кластеризации результатов прогнозирования с помощью алгоритмов

k-means и алгоритма понижения размерности – TSNE, позволяющего снизить размерность N-мерного пространства до заданного (в нашем случае 2D пространства) для отображения данных прогноза на плоскости.

В качестве системных показателей, характеризующих «здоровье компьютерной сети» используется уровень обслуживания (Service Level Objectives, SLO), и соглашение об уровне предоставляемого сервиса SLA (*Service Level Agreement*) рассматриваемой КС. Считается, что система нарушила цели уровня обслуживания, если введенный показатель превысил порог. Для исследуемой системы SLO было задано в следующем виде: задержка сигнала в сети (*signal_delay*) – менее 5 миллисекунд; время реакции сервера (*server_response_delay*) – менее 1.5 секунды; отброшенные сетевые пакеты (*network_packets_dropped*) – количество отброшенных входящих или исходящих сетевых пакетов больше 0; задержка дисковых операций (*disk_operation_delay*) – среднее время, затраченное на операции чтения или записи, превышает 2 секунды.

Если ни одна из целей уровня обслуживания не была нарушена, то состояние системы считается нормальным – *normal*.

В качестве атрибутов характеризующих состояния КС были выбраны: *server_response_tmetotal* – время ответа сервера на запрос; *network_outdropped* – количество отброшенных сетевых пакетов; *ping_avg* – среднее время ответа на запрос (при отправке трёх запросов); *network_inpackets* – количество полученных пакетов; *disk_iowriteawait* – среднее время запросов на запись, выданных на обслуживаемое устройство.

Результаты кластеризации представлены на рис. 3.

На рисунке 3а отображена принадлежность текущей точки прогноза к одному из кластеров, а также евклидово расстояние до центра каждого из пяти кластеров, характеризующих состояние КС. На рисунке 3 б в виде кластеров показано отображение текущего состояния КС на этапе обучения «в прошлом» и текущее состояние в виде точку (выделена черная точка) в текущем времени. На рисунке 3 в пока-

зано деление области состояний КС на кластеры, при условии, что каждая точка имеет отношение только к одному кластеру (как в прошлом, так и в настоящем). Количество кластеров в рассматриваемом случае равно 5.

Оценка «аномальности» КС

Количественно оценивать «здоровье сети» можно по близости текущей оценки прогноза i_j состояния КС в виде расстояния $\rho(p_s)$ до ближайших центров кластеров, помещенных символом p . Данные расстояния могут быть определены на порядковой шкале значениями от 1 до 5. Тогда степень критичности прогнозируемых аномальных состояний КС может быть описана соотношением

$$critical(i_j) = critical(p) \circ \min_s \rho(p_s), \quad (16)$$

где $critical(p) \in \{1, \dots, 5\}$ – степень критичности p – атрибут, определяемая экспертным путем; $\circ$ – операция на порядковых шкалах.

Порядковая шкала отражает более высокий уровень измерений, учитывающий, к какой категории принадлежит объект и в каком отношении он находится с другими объектами.

Общий уровень «здоровья» КС $C(u)$ может быть определен из соотношения

$$C(u) = \sum_{i=1}^k critical(i_i). \quad (17)$$

В результате решение об аномальности прогнозируемого состояния КС может приниматься по уровню $C(u)$. Уровень безопасности может быть представлен в цвете путем определения значения показателя «цвет инцидентов» ЦИ:

$$ЦИ = \begin{cases} \text{зеленый, если } C(u) \leq C^{\text{зеленый}}(u) \\ \text{желтый, если } C^{\text{зеленый}}(u) \leq C(u) < C^{\text{красный}}(u) \\ \text{красный, если } C(u) \geq C^{\text{красный}}(u) \end{cases} \quad (18)$$

Рис. 3. Результаты кластеризации результатов прогнозирования состояний КС

Здесь $C_{\text{зеленый}}(u)$ и $C_{\text{красный}}(u)$ пороговые уровни оцениваемые на этапе кластеризации. «Зеленый» уровень свидетельствует о отсутствии в прогнозе аномалии, что свидетельствует о нормальном режиме работы КС. «Красный» уровень свидетельствует о возможном появлении в прогнозе аномального состояния КС, является неприемлемым и требует немедленного реагирования. «Желтый» уровень аномальности не является приемлемым, однако не является и критичным.

Выводы

В процессе выполнения работы на основе мониторинга системных показателей, поведения пользователей и анализа системных журналов, а также прогнозирование появления аномальных состояний разработаны алгоритмы обнаружения и прогнозирования аномальных состояний, возникающих в КС. Предложена концепция формализации качества функционирования компьютерной сети под названием "Здоровье КС", а также разработаны алгоритмы, позволяющие выполнять диагностику и прогнозировать «здоровье КС» на основе последовательностных паттернов.

Методами интеллектуального анализа данных, на основе сбора и логирования значимой информации, разработаны методика и алгоритмы диагностики и прогнозирования «здоровья» компьютерной сети.

Выполнены численные оценки кластеризации состояний КС с целью определения характера нарушения «здоровья» сети.

Программный продукт, созданный на базе предложенных в рамках исследования алгоритмов, может быть интересен компаниям, занимающимся производством телекоммуникационного оборудования, эксплуатацией компьютерных сетей, а также операторам сетей передачи данных.

Литература

1. Xiaohui Gu. Online Anomaly Prediction for Robust Cluster Systems // IEEE 25th International Conference on Data Engineering. March 2009, pp. 1000-1011. DOI: 10.1109/ICDE.2009.128
2. Cohen, S. Zhang, M. Goldszmidt, J. Symons, T. Kelly, and A. Fox. Capturing, indexing, clustering, and retrieving system history. SOSP, 2005.
3. Mirza M., Sommers J., Barford P., Zhu J. A Machine Learning Approach to TCP Throughput Prediction // Proc. of ACM SIGMETRICS, 2007.
4. Шелухин О.И., Рябинин В.С., Фармаковский М.А. Обнаружение аномальных состояний компьютерных систем средствами интеллектуального анализа данных системных журналов // Вопросы кибербезопасности №2(26), 2018. DOI: 10.21681/2311-3456-2018-2-33-43
5. Шелухин О.И., Рябинин В.С. Обнаружение аномалий больших данных неструктурированных системных журналов // Вопросы кибербезопасности. 2019. №2(30). С. 36-41. DOI 10.21681/2311-3456-2019-2-36-41
6. Gniady C., Butt A.R., Hu Y.C. Program Counter Based Pattern Classification in Buffer Caching // Proc. of OSDI, 2004.
7. Mohammed J. Zaki. SPADE: An Efficient Algorithm for Mining Frequent Sequences // Machine Learning. №42. 2001. С. 31-60.
8. Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal and Mei-Chun Hsu. Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach // IEEE Transactions On knowledge and data engineering. Vol. 16. No. 10. 2004.
9. Srikant R., Agrawal R. Mining Sequential Patterns: Generalizations and Performance Improvements // EDBT, 1996.
10. Agrawal R., Srikant R. Mining sequential patterns // Proceedings of the Eleventh International Conference on Data Engineering, 1995.
11. Айвазян С.А., Енюков И.С., Мешалкин Л.Д. Прикладная статистика. Исследование зависимостей. М.: Финансы и статистика, 1985.
12. Abbasghorbani S., Tavoli R. Survey on sequential pattern mining algorithms // 2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran. 2015, pp. 1153-1164. doi: 10.1109/KBEI.2015.7436211
13. Philippe Fournier-Viger, Jerry Chun-Wei Lin, Rage-Uday Kiran, Yun-Sing Koh, and Rincy Thomas. 2017a. A survey of sequential pattern mining // Data Science and Pattern Recognition 1, 1 (2017), pp. 54-77.
14. Wensheng Gan, Jerry Chun-Wei Lin, Philippe Fournier-Viger, Han-Chieh Chao, and Philip S. Yu. A Survey of Parallel Sequential Pattern Mining // ACM Trans. Knowl. Discov. Data. 0, 1, Article 00 (August 2018). 33 p. <https://doi.org/0000001>
15. Jen-Wei Huang, Chi-Yao Tseng, Jian-Chih Ou, and Ming-Syan Chen. A General Model for Sequential Pattern Mining with a Progressive Database Publication // IEEE Transactions On Knowledge And Data Engineering. Vol. 20. No. 9. 2008.
16. Keshavamurthy B.N., Mitesh Shama, Durga Toshniwal. B. Efficient Support Coupled Frequent Pattern Mining Over Progressive Database Publication // International journal of Database Management Systems (IJDMS). Vol. 2. No.2.2010.
17. K.M.V. Madan Kumar, P.V.S. Srinivas and C. Raghavendra Rao. Sequential Pattern Mining With Multiple Minimum Supports in Progressive Databases Publication // International Journal of Database Management Systems (IJDMS). Vol. 4. No. 4. 2012.
18. Молодцов Д.А. Сравнение и продолжение многозначных зависимостей // Нечеткие системы и мягкие вычисления. 2016. Том 11, выпуск 2. С. 115-145.
19. Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И. Методы и модели анализа данных: OLAP и Data Mining. СПб.: БХВ-Петербург, 2004.

HEALTH MONITORING OF A COMPUTER NETWORK BASED ON SEQUENTIAL ANALYSIS OF SERIAL PATTERN

Oleg I. Sheluhin, MTUCI, Moscow, Russia, sheluhin@mail.ru

Andrey V. Osin, MTUCI, Moscow, Russia, osin_a_v@mail.ru

Denis V. Kostin, MTUCI, Moscow, Russia, d.v.kostin@mail.ru

Abstract

To solve the problems of monitoring the "health" of a computer network, it is proposed to use the methods and algorithms of statistical data processing, machine learning, and the intellectual analysis of historical data obtained by studying the behavior of the network in the past to extract patterns. We proposed to use the method of selecting the most appropriate pattern for "network health" to predict anomalous events in a computer network by using sequential analysis of extracted series of patterns. The stages of forecasting based on sequential patterns and the structure of the implemented algorithm for calculating the forecast of the current state of a computer system are considered in the article. It is proposed to use the Pareto function as a target function for optimizing the hyperparameters of the forecasting algorithm. The best set of parameters is determined by the maximum value of the target function. Service Level Objectives and Service Level Agreement are used as system indicators characterizing the "health of the computer network". A visualization of the clustering of states of a computer network is considered using a specific example using the k-means algorithm and the dimensional reduction algorithm TSNE. It is proposed to evaluate the "network health" as the distance of the forecast of the state of the computer network to the region of anomalous states formed as a result of clustering in the form of the distance to the nearest cluster centers, on an ordinal scale from 1 to 5. The paper proposes to evaluate the "health of the network using the "Green", "Yellow" and "Red" levels.

Keywords: anomaly states, computer network, forecasting, machine learning; data mining, monitoring system metrics, clustering, sequential analysis, pattern.

References

1. Xiaohui Gu. (2009). Online Anomaly Prediction for Robust Cluster Systems. *IEEE 25th International Conference on Data Engineering*. March 2009. Pages 1000-1011. DOI: 10.1109/ICDE.2009.128
2. Cohen, S. Zhang, M. Goldszmidt, J. Symons, T. Kelly, and A. Fox. (2005). Capturing, indexing, clustering, and retrieving system history. *SOSP*.
3. M. Mirza, J. Sommers, P. Barford, and J. Zhu. (2007). A Machine Learning Approach to TCP Throughput Prediction. *Proc. of ACM SIGMETRICS*.
4. Sheluhin O.I., Ryabinin V.S., Farmakovskiy M.A. (2018). Detection of abnormal conditions of computer systems by means of data mining system logs. *Cybersecurity Issues* No. 2 (26). DOI: 10.21681 / 2311-3456-2018-2-33-43
5. Sheluhin O.I., Ryabinin V.S. (2019). Detecting Big Data Anomalies in Unstructured System Logs. *Cybersecurity Issues*. No. 2 (30), pp. 36-41. DOI 10.21681 / 2311-3456-2019-2-36-41
6. C. Gniady, A. R. Butt, and Y. C. Hu. (2004). Program Counter Based Pattern Classification in Buffer Caching. *Proc. of OSDI*.
7. Mohammed J. Zaki. (2001). SPADE: An Efficient Algorithm for Mining Frequent Sequences. *Machine Learning*. No.42, pp. 31-60.
8. Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal and Mei-Chun Hsu. (2004). Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach. *IEEE Transactions On knowledge and data engineering*. Vol. 16. No. 10.
9. R. Srikant, R. Agrawal. (1996). Mining Sequential Patterns: Generalizations and Performance Improvements, EDBT.
10. R. Agrawal and R. Srikant. (1995). Mining sequential patterns. *Proceedings of the Eleventh International Conference on Data Engineering*.
11. Ayvazyan S.A., Enyukov I.S., Meshalkin L.D. (1985). *Applied statistics. Dependency research*. Moscow: Finance and statistics.
12. S. Abbasghorbani and R. Tavoli. (2015). Survey on sequential pattern mining algorithms. *2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, Tehran, pp. 1153-1164. doi: 10.1109/KBEI.2015.7436211
13. Philippe Fournier-Viger, Jerry Chun-Wei Lin, Rage-Uday Kiran, Yun-Sing Koh, and Rincy Thomas. (2017). A survey of sequential pattern mining. *Data Science and Pattern Recognition* 1, 1, pp. 54-77.
14. Wensheng Gan, Jerry Chun-Wei Lin, Philippe Fournier-Viger, Han-Chieh Chao and Philip S. Yu. (2018). A Survey of Parallel Sequential Pattern Mining. *ACM Trans. Knowl. Discov. Data*. 0, 1, Article 00 (August 2018), 33 p. <https://doi.org/00000001>
15. Jen-Wei Huang, Chi-Yao Tseng, Jian-Chih Ou, and Ming-Syan Chen. (2008). A General Model for Sequential Pattern Mining with a Progressive Database Publication. *IEEE Transactions On Knowledge And Data Engineering*. Vol. 20. No. 9.
16. Keshavamurthy B.N., Mitesh Sharma, Durga Toshniwal. B. (2010). Efficient Support Coupled Frequent Pattern Mining Over Progressive Database Publication. *International journal of Database Management Systems (IJDMS)*. Vol. 2. No.2.
17. K.M.V. Madan Kumar, P.V.S. Srinivas and C. Raghavendra Rao. (2012). Sequential Pattern Mining With Multiple Minimum Supports in Progressive Databases Publication. *International Journal of Database Management Systems (IJDMS)*. Vol. 4. No. 4.
18. D. A. Molodtsov. (2016). Comparison and continuation of multi-valued dependencies. *Fuzzy Systems and Soft Computing*. Vol. 11, Issue 2, pp. 115-145.
19. Barseghyan A.A., Kupriyanov M.C., Stepanenko B.B., Kholod I.I. (2004). *Methods and models of data analysis: OLAP and Data Mining*. SPb.: BH St. Petersburg.

Information about authors:

Oleg I. Sheluhin, doctor of technical sciences, professor, head of the Department of Information Security, MTUCI, Moscow, Russia

Andey V. Osin, PhD, MTUCI, Moscow, Russia

Denis V. Kostin, graduate student, MTUCI, department of information security, Moscow, Russia