DOI: 10.55648/1998-6920-2024-18-1-3-15 УДК 519.237:004.492

# Морфологический метод обнаружения аномальных состояний сервера

А. Д. Петров[1,2], Е. А. Харченко[1]

[1] Московский политехнический университет
[2] ООО «Безопасная информационная зона»

*Аннотация*: В работе предложен вычислительно простой алгоритм выявления выбросов и аномалий на основе морфологического анализа внутренней структуры многомерных данных. Важным преимуществом метода является возможность одновременной работы как с качественными, так и с количественными признаками. От аналогов его также отличает простота представления и интерпретации результатов. Доверительная область значений изучаемых объектов аппроксимируется объединением доверительных областей значений качественно однородных объектов (кластеров). Принадлежность объектов одному кластеру обуславливается характерными для предметной области причинно-следственными связями между признаками. В основе метода лежит построение конечного вероятностного пространства, каждый элемент которого (двоичный вектор) однозначно ставится в соответствие объектам выборки. На основании неравенства Чебышёва за выбросы принимаются маломощные кластеры. За аномалии принимаются объекты, не принадлежащие совокупной доверительной области. Проработаны основанные на расстоянии Хэмминга механизмы сравнения: 1) кластера и кластера; 2) кластера и объекта; 3) объекта и объекта. Для демонстрации действенности метода разработан программный модуль для обнаружения аномальных состояний сервера на базе операционной системы семейства Linux. Он также может быть использован в качестве вспомогательного в профессиональных системах обнаружения вторжений.

*Ключевые слова*: многомерные данные, выбросы, аномалии, кластеризация, сегментация, машинное обучение без учителя, системы обнаружения вторжений.

*Для цитирования*: Петров А. Д., Харченко Е. А. Морфологический метод обнаружения аномальных состояний сервера // Вестник СибГУТИ. 2024. Т. 18, № 1. С. 3–15. https://doi.org/10.55648/1998-6920-2024-18-1-3-15.



Статья поступила в редакцию 05.07.2023;
принята к публикации 12.08.2023.

## 1. Введение

Фокус внимания в области кибербезопасности неуклонно смещается в сторону разработки специализированных средств защиты информационных систем от целевых атак и сложных угроз, в том числе и комплексных таргетированных угроз [1]. В отличие от типового вредоносного программного обеспечения, целевые атаки осуществляются под активным контролем и удалённым управлением мотивированных и квалифицированных злоумышленников. Их сложно выявить, т.к. они всегда многоэтапные, при этом каждый отдельный шаг злоумышленника

в защищаемой системе может выглядеть легитимно. Наибольшую опасность из-за масштабности последствий представляют целевые атаки на объекты критической информационной инфраструктуры.

Большинство специализированных средств обнаружения целевых атак являются пассивными, поскольку не должны нарушать непрерывность производственных процессов: недопустимо прерывание работы основного программного обеспечения (в результате значительного потребления ресурсов защищаемой системы) или оперативная блокировка процессов (в результате ложных срабатываний). Такие решения не следят за поведением приложений или процессов в режиме реального времени, а изучают оставленные в системных журналах следы (логи) программ, задействуя при этом стандартные интерфейсы эксплуатируемой операционной системы.

Существуют два принципиально различных подхода к выявлению несанкционированной и вредоносной активности в компьютерной сети или на отдельном хосте: 1) на основании сигнатур (ранее известных способов проникновения) – проверяемые данные сравниваются с известными образцами сигнатур атаки, и в случае их совпадения создаётся оповещение безопасности; 2) на основании аномалий (новых угроз) – активность в сети или на хосте сравнивается с моделью корректного, доверенного поведения контролируемых элементов и фиксирует отклонения от неё.

Сложность самой природы предметной области обуславливает применение в целях детектирования аномальных состояний серверного оборудования методов машинного обучения. Наибольшую точность показывают модели, построенные на нейронных сетях [2], использование которых не лишено очевидных недостатков: невозможность интерпретации результатов и ресурсоёмкость вычислительных процессов. При использовании же базовых алгоритмов машинного обучения [3, 4, 5, 6] не исследуется или искажается внутренняя структура исходных данных. Приведём пример.

Обучающие выборки (датасеты) являются многомерными и в общем случае имеют неоднородную структуру. Но зачастую неразмеченные данные или данные одного класса рассматриваются как объекты одной сущности, каждый атрибут которой имеет нормальное распределение (его значения усредняются). Согласно первичному определению, за аномалии принимаются объекты, лежащие за границами доверительной области. Очевидно, что основной недостаток такого подхода – повышение порога детектирования аномалий (рис. 1, слева).

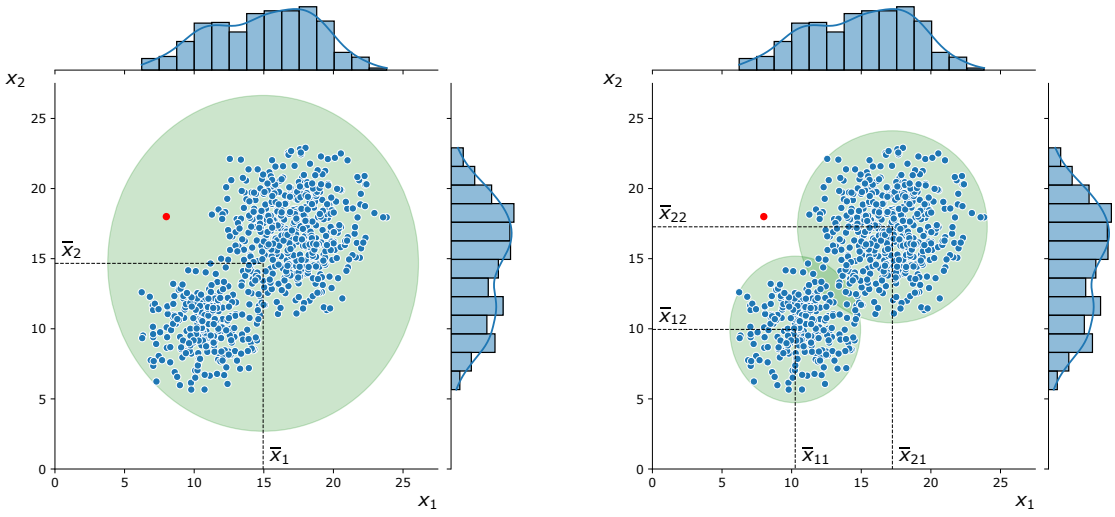


Рис. 1. Аномалия на фоне несегментированных (слева) и сегментированных (справа) данных

Кроме того, признаки датасета часто рассматриваются как независимые, что не может объективно отражать специфику предметной области. Так, у аномального объекта каждый признак

по отдельности вполне может попадать в свою область допустимых значений, в то время как в совокупности значения всех признаков не будут допустимыми (рис. 1, справа). Концепция современных решений мониторинга информационных процессов состоит в том, что каждое событие изучается не обособленно, а в контексте остальных событий.

В настоящей работе предложен один универсальный метод выявления выбросов и детектирования аномалий, учитывающий морфологию многомерных данных. Он прост для понимания и использования, т.к. построен на базовых понятиях линейной алгебры, теории вероятностей и статистики, что немаловажно для решения практических задач информационной безопасности. Также в рамках работы выработана реляционная модель состояния сервера и способ наполнения её данными.

Для подтверждения действенности метода и адекватности модели приводятся результаты работы программного модуля, предназначенного для пассивного обнаружения аномальных и подозрительных состояний сервера на базе операционной системы семейства Linux. Выбор операционной системы продиктован последними изменениями законодательства, согласно которым госорганам и госзаказчикам на критической инфраструктуре с 1 января 2025 года запрещается использовать иностранное программное обеспечение [7]. Наиболее распространёнными отечественными операционными системами в настоящее время являются Astra Linux и ALT Linux.

## 2. Морфологический метод выявления аномалий в многомерных данных

Предлагаемый метод выявления аномалий извлечён из метода принятия управленческих решений, освещённого в работах [8, 9], и по сути представляет собой метод кластеризации многомерных данных. В его основе лежит построение конечного вероятностного пространства $L$, множеством элементарных событий которого является линейное пространство двоичных векторов, в котором определены следующие операции:

1) сложение векторов – операция над двумя векторами, результатом которой является вектор, каждая составляющая которого равна сумме по модулю два одноимённых составляющих исходных векторов:

$$\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_k \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \end{pmatrix} = \begin{pmatrix} (a_1 + b_1) \bmod 2 \\ (a_2 + b_2) \bmod 2 \\ \vdots \\ (a_k + b_k) \bmod 2 \end{pmatrix};$$

2) умножение вектора на скаляр – операция, результатом которой является вектор, составляющие которого равны конъюнкции скаляра и одноимённой составляющей исходного вектора:

$$\lambda \cdot \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_k \end{pmatrix} = \begin{pmatrix} \lambda \,\&\, a_1 \\ \lambda \,\&\, a_2 \\ \vdots \\ \lambda \,\&\, a_k \end{pmatrix}.$$

Рассмотрим этапы построения пространства $L$.

Область допустимых значений $X_j$ каждого показателя $x_j$ разбивается на непересекающиеся подмножества, каждому из которых $X_{\alpha j}$ взаимно однозначно соответствует своя составляющая (свой бит) произвольного вектора пространства $L$ (рис. 2):

$$X_j = X_{1j} \cup X_{2j} \cup \ldots \cup X_{k_j j},$$

причём $X_{\alpha j} \cap X_{\beta j} \neq 0$ тогда и только тогда, когда $X_{\alpha j} = X_{\beta j}$. Здесь $k_j$ – число качественно однородных подмножеств области допустимых значений показателя $x_j$. Размерность пространства $L$ равна:

$$k = k_1 + k_2 + ... + k_n,$$

где $n$ – число показателей.

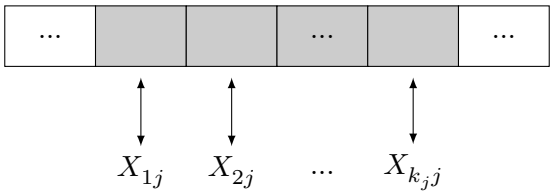


Рис. 2. Группа компонент вектора $\bar{y}_i \in L$, соответствующая показателю $x_j$

В случае, когда показатель $x_j$ является качественным (измеренным в номинальной или ранговой шкале), т.е.

$$x_j \in \{x_{1j}, x_{2j}, ..., x_{k_j j}\},$$

где $x_{\alpha j}$ – $\alpha$-ое значение показателя $x_j$, в рассмотрение вводятся одноэлементные непересекающиеся множества

$$X_{\alpha j} = \{x_{\alpha j}\}.$$

Каждая область $X_{\alpha j}$ представляется единственным входящим в неё элементом.

В случае, когда показатель $x_j$ является количественным (измеренным в шкале интервалов или отношений), строится сглаженная статистическая плотность распределения $\varphi(x_j)$ значений случайной величины $x_j$, в общем случае она имеет многовершинный характер (рис. 3).

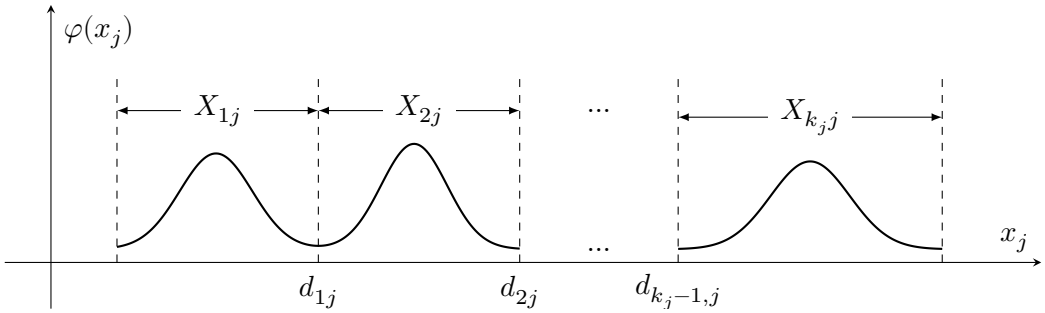


Рис. 3. Область допустимых значений $X_j$ количественного показателя $x_j$

Точки минимумов $d_{\alpha j}$ плотности распределения $\varphi(x_j)$ случайной величины $x_j$ разбивают область допустимых значений $X_j$ количественного показателя $x_j$ на непересекающиеся подмножества:

$$\begin{cases} X_{1j} = \{x_j \mid x_j < d_{1j}\}, \\ X_{2j} = \{x_j \mid d_{1j} \leqslant x_j < d_{2j}\}, \\ ... \\ X_{k_j j} = \{x_j \mid x_j \geqslant d_{k_j-1,j}\}. \end{cases}$$

Каждая область сгущения значений $X_{\alpha j}$ имеет единственную вершину плотности распределения $\varphi(x_j)$ и представляется условными математическим ожиданием и средним квадратическим отклонением показателя $x_j$.

Разбиению одномерных областей допустимых значений всех показателей на непересекающиеся подмножества соответствует разбиение многомерной области допустимых значений $X$ объектов моделируемой сущности (в нашем случае – нормального состояния сервера), т.е. различных комбинаций значений показателей, на непересекающиеся подмножества (кластеры), каждое из которых либо вообще не содержит ни одной комбинации значений показателей, либо содержит только такие комбинации значений показателей, у которых значения одноимённых качественных показателей равны, а значения каждого количественного показателя принадлежат одному и тому же подмножеству разбиения области допустимых значений этого показателя.

Таким образом:

$$X = \bigcup_{\substack{\alpha_1 \in \overline{1,k_1},\\ \dots \\ \alpha_n \in \overline{1,k_n}}} (X_{\alpha_1 1} \times X_{\alpha_2 2} \times ... \times X_{\alpha_n n}),$$

где

$$X_{\alpha_1 1} \times X_{\alpha_2 2} \times ... \times X_{\alpha_n n} =$$
$$= \{(x_1, x_2, ..., x_n) \mid (x_1 \in X_{\alpha_1 1}) \,\&\, (x_2 \in X_{\alpha_2 2}) \,\&\, ... \,\&\, (x_n \in X_{\alpha_n n})\},$$

причём $(X_{\alpha_1 1} \times X_{\alpha_2 2} \times ... \times X_{\alpha_n n}) \cap (X_{\beta_1 1} \times X_{\beta_2 2} \times ... \times X_{\beta_n n}) \neq 0$ тогда и только тогда, когда $\alpha_i = \beta_i$, $i \in \overline{1,n}$.

Рассмотрим множество $C$, элементами которого являются подмножества рассмотренного выше разбиения области допустимых значений моделируемой сущности:

$$C = \{(X_{\alpha_1 1}, X_{\alpha_2 2}, ..., X_{\alpha_n n}) \mid (X_{\alpha_1 1} \in C_1) \,\&\, (X_{\alpha_2 2} \in C_2) \,\&\, ... \,\&\, (X_{\alpha_n n} \in C_n)\},$$

где

$$\begin{cases} C_1 = \{X_{11}, X_{21}, ..., X_{k_1 1}\}, \\ C_2 = \{X_{12}, X_{22}, ..., X_{k_2 2}\}, \\ ... \\ C_n = \{X_{1n}, X_{2n}, ..., X_{k_n n}\}. \end{cases}$$

Каждому элементу множества $C$ однозначно соответствует свой вектор введённого линейного пространства. Из способа построения этого пространства следует, что в данном векторе в группе составляющих (битов), биективно соответствующей показателю $x_j$, существует единственная $1$, остальные разряды этой группы равны $0$. Векторам, у которых хотя бы в одной из рассматриваемых групп разрядов отсутствует $1$ либо их число больше одной, не соответствуют элементы множества $C$. Векторам, у которых в каждой из рассматриваемых групп разрядов существует единственная $1$, взаимно однозначно соответствуют элементы множества $C$.

В пространстве $L$ определено расстояние Хэмминга между двумя произвольными векторами:

$$\rho_{\text{ham}}(\bar{y}_i, \bar{y}_j) = \rho_{\text{ham}}((y_{1i}, y_{2i}, ..., y_{ki}), (y_{1j}, y_{2j}, ..., y_{kj})) =$$
$$= |\{(y_{\alpha i}, y_{\alpha j}) \mid y_{\alpha i} + y_{\alpha j} = 1\}|.$$

Расстояние между двумя элементами множества $C$ определим как половину расстояния Хэмминга между соответствующими векторами пространства $L$:

$$\rho((X_{\alpha_1 1}, X_{\alpha_2 2}, ..., X_{\alpha_n n}), (X_{\beta_1 1}, X_{\beta_2 2}, ..., X_{\beta_n n})) = \frac{1}{2} \cdot \rho_{\text{ham}}(\bar{y}_i, \bar{y}_j). \quad (1)$$

Непосредственной подстановкой можно доказать, что половина расстояния Хэмминга удовлетворяет аксиомам расстояния. Из способа построения пространства $L$ следует, что это расстояние численно равно количеству пар одноимённых составляющих $(X_{\alpha_j j}, X_{\beta_j j})$, у которых $X_{\alpha_j j} \neq X_{\beta_j j}$.

Расстояние между двумя комбинациями значений показателей сущности (т.е. между двумя объектами сущности) положим равным расстоянию между двумя элементами множества $C$ (т.е. между двумя кластерами), которым эти комбинации значений принадлежат. Таким образом, расстояние между двумя комбинациями значений показателей равно числу пар одноимённых показателей, которые принадлежат различным подмножествам из разбиения области допустимых значений их комбинаций.

Расстояние между отдельной комбинацией значений показателей (объектом сущности) и произвольным элементом множества $C$ (кластером) положим равным расстоянию между двумя соответствующими векторами пространства $C$.

Элементарным событиям (векторам введённого пространства $L$) припишем вероятности следующим образом: если вектор не соответствует ни одному элементу множества $C$ (формальным критерием этого является отсутствие единиц или наличие более одной единицы в группе разрядов вектора, соответствующего хотя бы одному из показателей), ему приписывается вероятность, равная $0$; если вектор соответствует элементу множества $C$ (формальным критерием этого является наличие единственной единицы в каждой группе разрядов вектора, соответствующей одному из показателей), ему приписывается вероятность, равная отношению числа объектов сущности, соответствующих рассматриваемому элементу множества $C$, к общему числу $m$ объектов сущности.

Во многих практических задачах кластеризации объекты сущности будут группироваться в небольшом числе элементов множества $C$. Большинство же других элементов множества $C$ будут иметь нулевую или близкую к нулю вероятность.

Упорядочим элементарные события (двоичные вектора) в соответствии с убыванием их вероятностей. После этого введём новую дискретную случайную величину $\xi$, значения которой равны номерам элементарных событий в этом упорядоченном ряду, а вероятности – вероятностям соответствующих элементарных событий.

После вычисления математического ожидания $\mu_\xi$ и среднего квадратического отклонения $\sigma_\xi$ этой случайной величины на основании неравенства Чебышёва

$$P(|\xi - \mu_\xi| \geqslant 3\sigma_\xi) \leqslant \frac{1}{9}$$

можно заключить, что подавляющее большинство объектов моделируемой сущности (не меньше 89 %) принадлежит тем элементам множества $C$, которым соответствуют значения случайной величины $\xi$, удовлетворяющие неравенству

$$1 \leqslant \xi \leqslant \lfloor \mu_\xi + 3\sigma_\xi \rfloor, \tag{2}$$

и доля выбросов не превышает 11 %.

Тогда за доверительную область объектов исследуемой сущности следует принять объединение доверительных областей только таких элементов множества $C$, которые соответствуют практически возможным значениям величины $\xi$. Детектирование аномального объекта, в свою очередь, сводится к проверке объекта на непринадлежность совокупной доверительной области сущности.

Доверительная область объектов отдельного элемента множества $C$ представляет собой многомерный эллипсоид, описываемый уравнением:

$$\frac{(x_1 - \mu_{x_1})^2}{(3\sigma_{x_1})^2} + \frac{(x_2 - \mu_{x_2})^2}{(3\sigma_{x_2})^2} + ... + \frac{(x_n - \mu_{x_n})^2}{(3\sigma_{x_n})^2} = 1, \tag{3}$$

положение и полуоси эллипса определяются средними значениями и средними квадратическими отклонениями показателей объектов, принадлежащих рассматриваемому кластеру (здесь законы распределения показателей полагаем неизвестными).

## 3. Реляционная модель состояния сервера

При эксплуатации любой компьютерной системы продуцируется большое число так называемых сигналов (значения сенсоров, команд, параметров логики управления и т.д.), они тесно взаимосвязаны, что определяется физикой и логикой производственных процессов. Вследствие этого воздействие на одни параметры процесса неизбежно влечёт за собой изменение других параметров. В совокупности показания «в моменте» всех источников сигналов системы (условных сенсоров) определяют её состояние – нормальное (штатное) или аномальное (потенциально опасное).

Формализовать характер связей между показаниями сенсоров в общем случае не представляется возможным, поэтому на практике по собранным показаниям сенсоров формируется реляционная модель нормального состояния системы (с неявным учетом корреляций между сигналами). Тогда под аномальными понимают нетипичные состояния системы (значительно удалённые от нормальных).

В данной работе для формирования обучающей выборки, описывающей состояние сервера, разработана следующая схема. Предварительно регистрируются характеристики запущенных процессов:

- `pid` – идентификатор процесса;
- `name` – название процесса;
- `username` – пользователь, от имени которого запущен процесс;
- `ppid` – идентификатор родителя процесса;
- `parent_name` – название родителя процесса;
- `cpu_percent` – процент потребления ресурсов процессора процессом;
- `memory_percent` – процент потребления ресурсов оперативной памяти процессом;
- `num_threads` – количество потоков процесса;
- `terminal` – идентификатор терминала, из которого был запущен процесс;
- `nice` – приоритет выполнения процесса;
- `cmdline` – команда, которой был запущен процесс;
- `exe` – путь к исполняемому файлу процесса;
- `status` – статус процесса;
- `create_time` – время запуска процесса;
- `connections` – количество открытых соединений процесса;
- `open_files` – количество открытых файлов процесса.

Данные о процессах собираются с некоторым настраиваемым шагом (в работе равным одной минуте) с помощью кроссплатформенной библиотеки psutil языка программирования Python.

Обучающий датасет производится агрегированием данных исходного датасета по следующему правилу: для каждой строки с временем $t_c$ агрегируются все строки с временем $t_r$, которые отвечают условию

$$(t - \Delta) < t_r \leqslant t_c, \tag{4}$$

где $\Delta$ – некоторый небольшой заданный промежуток времени (в работе равный десяти минутам). Для того, чтобы процесс не учитывался несколько раз во время агрегации, все процессы, попадающие в заданный интервал перед основной агрегацией, группируются по идентификаторам процесса (`pid`) и по пользователям (`username`), от имени которых запущены процессы. При этом все количественные характеристики усредняются, а качественные соединяются в строку через запятую.

В результате методом скользящего окна была выработана следующая система суррогатных признаков для моделирования состояния сервера:

- `cpu_percent_avg` – средний процент потребления ресурсов процессора;

– memory_percent_avg – средний процент потребления ресурсов оперативной памяти;
– num_threads_avg – среднее количество потоков;
– connections_avg – среднее количество открытых соединений;
– open_files_avg – среднее количество открытых файлов;
– cpu_percent_sum – сумма процентов потребления ресурсов процессора;
– memory_percent_sum – сумма процентов потребления ресурсов оперативной памяти;
– num_threads_sum – сумма количества потоков;
– connections_sum – сумма количества открытых соединений;
– open_files_sum – сумма количества открытых файлов;
– cpu_percent_max – максимальный процент потребления ресурсов процессора;
– memory_percent_max – максимальный процент потребления ресурсов оперативной памяти;
– num_threads_max – максимальное количество потоков;
– connections_max – максимальное количество открытых соединений;
– open_files_max – максимальное количество открытых файлов;
– idle_status_count – количество процессов со статусом idle;
– sleeping_status_count – количество процессов со статусом sleeping;
– running_status_count – количество процессов со статусом running;
– zombie_status_count – количество процессов со статусом zombie;
– disk_sleep_status_count – количество процессов со статусом disk_sleep;
– root_processes_count – количество процессов, запущенных от имени суперпользователя (root);
– system_processes_count – количество системных процессов (имя пользователя начинается с sys);
– time_of_day – время (ночь, утро, день, вечер).

Признак автоматически исключается из датасета, если не является «говорящим», т.е. имеет одно значение (для качественного признака) или одну область сгущения значений (для количественного признака), – по нему объекты сущностно неразличимы. Для наглядности на рис. 4 приведены распределения значений признаков расчётного примера.

## 4. Программный модуль выявления аномальных состояний сервера

Формализованный выше математический аппарат для выявления выбросов и аномалий в многомерных данных лежит в основе разработанного в рамках работы программного модуля детектирования аномальных состояний сервера на базе операционной системы семейства Linux. Для демонстрации работы модуля был выбран сервер на базе операционной системы Ubuntu 20.04 со следующими характеристики: процессор Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60 GHz, ОЗУ 4 Гб.

На сервере был запущен Django-прект с использованием docker-compose, вместе с ним в Docker-контейнерах были развёрнуты база данных на основе СУБД PostgreSQL и web-сервер nginx. Также на сервере был запущен внешний nginx, который перенаправлял запросы Django-приложению и клиенту детектора. Для запуска сборщика данных, детектора аномалий и клиента в фоновом режиме использовался сервис supervisor.

Сбор первичных данных осуществлялся в течение четырёх дней. За весь период сервер использовался в обычном режиме в контролируемых условиях. Никаких сбоев и отказов не наблюдалось, поэтому собранные данные можно считать образцовыми. За всё время было собрано 121 МБ данных, или 889039 записей с информацией о запущенных процессах. Записанные в CSV-файл данные были выгружены с сервера и импортированы в локальную базу данных.

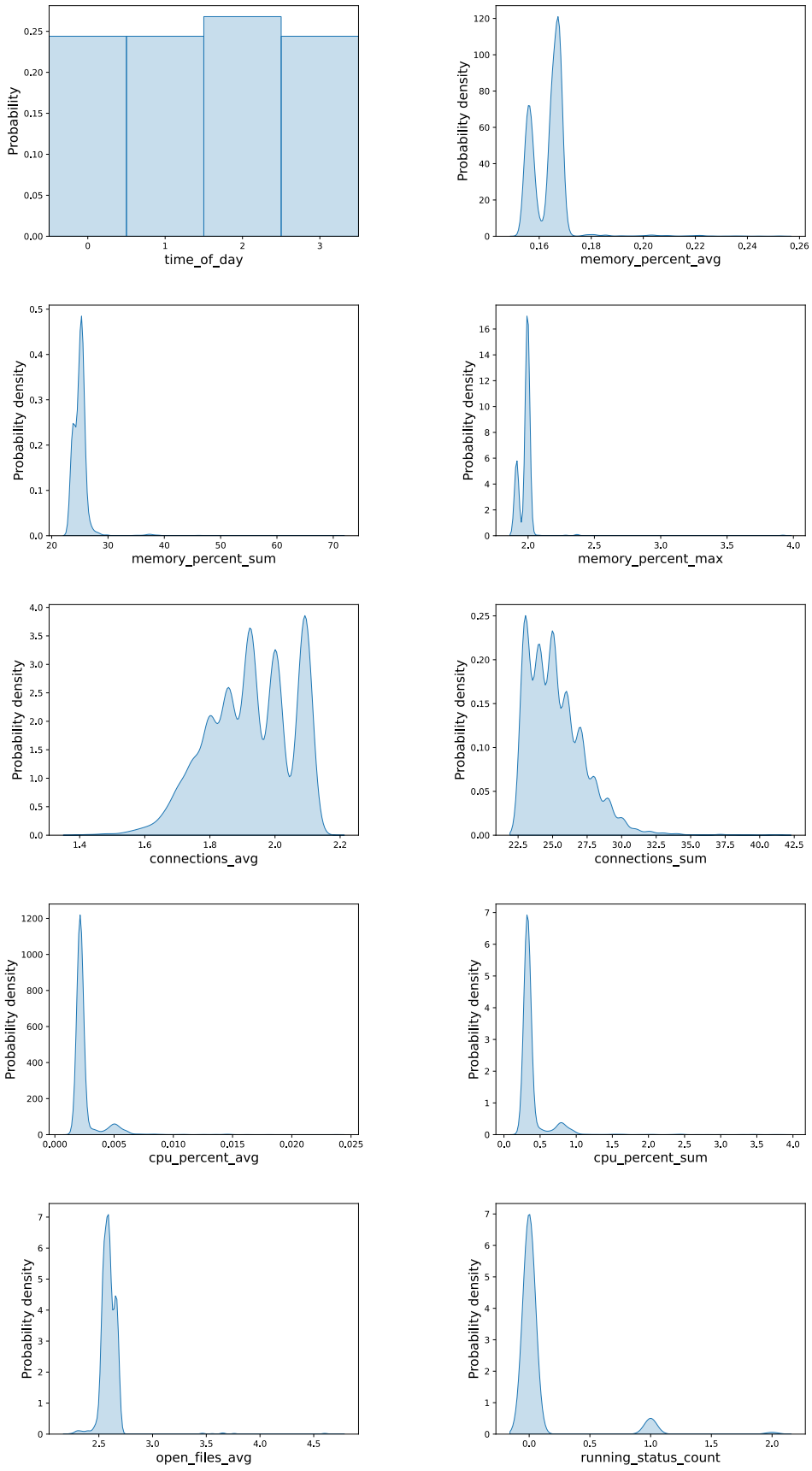


Рис. 4. Распределение значений признаков расчётной обучающей выборки

Затем они были поданы на вход детектора аномалий, причём без традиционной нормализации данных. После агрегации по правилу (4) был получен обучающий датасет из десяти признаков (их структура приведена на рис. 4). В процессе обучения на основании (2) детектором автоматически было выделено 313 кластеров нормальных состояний сервера (и отброшены выбросы – маломощные кластеры).

Размерность двоичных векторов, отождествляемых с кластерами, составила 32. Под детализацию области допустимых значений признака `time_of_day` программой было выделено 4 бита, признака `memory_percent_avg` – 2 бита, признака `memory_percent_sum` – 2 бита, признака `memory_percent_max` – 2 бита, признака `connections_avg` – 5 бит, признака `connections_sum` – 8 бит, признака `cpu_percent_avg` – 2 бита, признака `cpu_percent_sum` – 2 бита, признака `open_files_avg` – 3 бита и признака `running_status_count` – 2 бита.

Так, например, наибольшую частоту показал кластер, морфология которого представляется вектором 1000 01 01 01 00001 10000000 10 10 010 10. Здесь единица во второй группе битов трактуется как принадлежность значения признака `memory_percent_avg` интервалу, представляемому $\mu_{22} = 16.77\text{e-}2$ и $\sigma_{22} = 66.58\text{e-}5$.

Для проверки корректности работы детектора был проведён ряд экспериментов:

1. Обычная работа с сервером: посетили несколько раз развёрнутый на сервере сайт, выполнили несколько действий на сайте; зашли на сервер, посмотрели логи приложений, переключились между директориями.
2. Запуск процесса, который раньше никогда не запускался: запустили предварительно написанный на языке Python скрипт, который сгенерировал в большом объёме случайные данные, записали их во временную директорию `/tmp`, затем считали и удалили созданные файлы.
3. Подбор логина и пароля для ssh: скачали список из пятисот самых распространённых паролей, создали файл с распространёнными логинами пользователей и с помощью утилиты hydra запустили команду подбора пароля.
4. Реализация DoS-атаки на сервер: с помощью программы Cyphon-DoS послали большое количество запросов к серверу.

Детектор не выявил аномальные состояния при нормальной работе сервера и, наоборот, чётко детектировал аномальные состояния, которые проявились в процессе смоделированных компьютерных атак. Для повышения чувствительности алгоритма был установлен порог детектирования аномалии, равный на основании (1) одному отличающемуся признаку.

На более сложных данных детектор ожидаемо показывает более низкую эффективность. Это объясняется избыточностью доверительных областей кластеров (в пользу иллюстративности метода). Границы и ориентацию доверительных областей можно уточнить путем уменьшения числа стандартных отклонений в (3) или применения метода главных компонент к каждому кластеру.

## 5. Заключение

Одна из современных парадигм информационной безопасности заключается в том, что невозможно гарантированно предотвратить проникновение в систему, но важно как можно быстрее обнаружить подозрительное поведение системы и не позволить атаке развиться до наступления недопустимого события. С этой целью разрабатывают специальные, требующие корректного использования, инструменты обнаружения аномального или подозрительного поведения системы и оповещения о нём.

В настоящей работе предложен вычислительно простой алгоритм выявления выбросов и аномалий на основе морфологического анализа внутренней структуры многомерных данных.

Важным преимуществом метода является возможность одновременной работы как с качественными, так и с количественными признаками. От аналогов его также отличает простота представления и интерпретации результатов. По сути, доверительная область изучаемых объектов аппроксимируется объединением доверительных областей качественно однородных объектов.

Разработанный на основе представленного метода программный модуль может быть использован в качестве вспомогательного в полноценных IDS-системах. Как правило, они чувствительны и к несанкционированным сбоям или отказам системы, которые также несут потенциальные угрозы информационной безопасности.

## Литература


1. *Левцов В.* Анатомия таргетированной атаки [Электронный ресурс]. URL: https://www.kaspersky.ru/blog/targeted-attack-anatomy/4388 (дата обращения: 28.06.2023).
2. *Лаврентьев А.* MLAD: обнаружение аномалий методами машинного обучения [Электронный ресурс]. URL: https://ics-cert.kaspersky.ru/publications/reports/2018/01/16/mlad-machine- learning-for-anomaly-detection (дата обращения: 28.06.2023).
3. Указ Президента Российской Федерации от 30.03.2022 № 166 ”О мерах по обеспечению технологической независимости и безопасности критической информационной инфраструктуры Российской Федерации”[Электронный ресурс]. URL: http://publication.pravo.gov.ru/Document/View/0001202203300001 (дата обращения: 28.06.2023).
4. ГОСТ Р ИСО 16269-4-2017 ”Статистические методы. Статистическое представление данных. Часть 4. Выявление и обработка выбросов”. М.: Стандартинформ, 2017. 53 с.
5. *Дьяконов А. Г., Головина А. М.* Выявление аномалий в работе механизмов методами машинного обучения // Аналитика и управление данными в областях с интенсивным использованием данных. 2017. С. 469–476.
6. *Han J., Kamber M., Pei J.* Data Mining: Concepts and Techniques. Morgan Kaufmann, 2011. 740 p.
7. *Tan P.-N., Steinbach M., Karpatne A., Kumar V.* Introduction to Data Mining. Pearson, 2019. 839 p.
8. *Харченко Е. А.* Морфологический подход к принятию обоснованных решений по экспертным суждениям // Вестник ТвГУ. Серия: Прикладная математика. 2019. № 2. С. 42–56. https://doi.org/10.26456/vtpmk531.
9. *Харченко Е. А.* Алгоритм морфологического метода экспертных оценок для решения задачи прогнозирования // Компьютерные инструменты в образовании. 2023. № 2. С. 5–20. https://doi.org/10.32603/2071-2340-2023-2-5-20.



**Петров Антон Денисович**

магистрант кафедры информационной безопасности Московского политехнического университета (Московский Политех, 107023, Москва, ул. Большая Семёновская, д. 38); разработчик направления анализа защищённости ООО «Безопасная информационная зона» (ООО «БИЗон», 105066, Москва, ул. Ольховская, д. 4, корп. 2), e-mail: antonp2@yandex.ru, ORCID ID: 0009-0007-9546-8544.

**Харченко Елена Алексеевна**
старший преподаватель кафедры инфокогнитивных технологий Московского политехнического университета (Московский Политех, 107023, Москва, ул. Большая Семёновская, д. 38), e-mail: elenakhaa@yandex.ru, ORCID ID: 0000-0002-5082-4564.

*Авторы прочитали и одобрили окончательный вариант рукописи.*
*Авторы заявляют об отсутствии конфликта интересов.*
Вклад соавторов: *каждый автор внёс равную долю участия как во все этапы проводимого теоретического исследования, так и при написании разделов данной статьи.*

## Morphological Method for Detecting Abnormal Server States

**Anton D. Petrov**[1,2], **Elena A. Kharchenko**[1]

[1] Moscow Polytechnic University,
[2] «BiZone» Limited Liability Company

*Abstract*: The paper proposes a computationally simple algorithm for detecting outliers and anomalies based on morphological analysis of the internal structure of multidimensional data. An important advantage of the method is the possibility of simultaneous work with qualitative and quantitative signs. It is also distinguished from its analogues by the simplicity of presentation and interpretation of the results. The values' confidence range of the studied objects is approximated by combining the values' confidence ranges of qualitatively homogeneous objects (clusters). The belonging of objects to one cluster is determined by the causal relationships between the features characteristic of the subject area. The method is based on the construction of a finite probability space and each element of binary vector is uniquely assigned to the objects of the sample. Based on the Chebyshev inequality, low-power clusters are taken as emissions. Objects that do not belong to the aggregate confidence area are taken as anomalies. Comparison mechanisms based on the Hamming distance have developed: 1) cluster and cluster; 2) cluster and object; 3) object and object. To demonstrate the effectiveness of the method a software module for detecting abnormal server states based on the Linux operating system has been developed. It can also be used as an auxiliary in professional intrusion detection systems.

*Keywords*: multidimensional data, outliers, anomaly, clustering, segmentation, unsupervised learning, intrusion detection system.

*For citation*: Petrov A. D., Kharchenko E. A. Morphological method for detecting abnormal server states (in Russian). *Vestnik SibGUTI*, 2024, vol. 18, no. 1, pp. 3-15. https://doi.org/10.55648/1998-6920-2024-18-1-3-15.



The article was submitted: 05.07.2023;
accepted for publication 12.08.2023.

### References

1. Levtsov V. Anatomiya targetirovannoj ataki [The anatomy of a targeted attack], available at: https://www.kaspersky.ru/blog/targeted-attack-anatomy/4388 (accessed 28.06.2023).

2. Lavrentyev A. MLAD: obnaruzhenie anomalij metodami mashinnogo obucheniya [MLAD: Anomaly detection by machine learning methods], available at: https://ics-cert.kaspersky.ru/publications/reports/2018/01/16/mlad-machine- learning-for-anomaly-detection (accessed 28.06.2023).
3. Ukaz Prezidenta Rossijskoj Federacii ot 30.03.2022 № 166 ”O merah po obespecheniyu tekhnologicheskoj nezavisimosti i bezopasnosti kriticheskoj informacionnoj infrastruktury Rossijskoj Federacii” [Decree of the President of the Russian Federation No. 166 dated 30.03.2022 ”On Measures to ensure the Technological Independence and security of the Critical Information Infrastructure of the Russian Federation”], available at: http://publication.pravo.gov.ru/Document/View/0001202203300001 (accessed 28.06.2023).
4. *GOST R ISO 16269-4-2017 ”Statisticheskie metody. Statisticheskoe predstavlenie dannyh. CHast’ 4. Vyyavlenie i obrabotka vybrosovi”* [ISO 16269-4-2017 ”Statistical methods. Statistical data presentation. Part 4. Detection and treatment of outliers”]. Moscow, Standartinform, 2017. 53 p.
5. D’yakonov A. G., Golovina A. M. Vyyavlenie anomalij v rabote mekhanizmov metodami mashinnogo obucheniya [Anomaly detection in mechanisms using machine learning]. *Analitika i upravlenie dannymi v oblastyah s intensivnym ispol’zovaniem dannyh*, 2017, pp. 469-476.
6. Han J., Kamber M., Pei J. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2011. 740 p.
7. Tan P.-N., Steinbach M., Karpatne A., Kumar V. *Introduction to Data Mining*. Pearson, 2019. 839 p.
8. Kharchenko E. A. Morfologicheskij podhod k prinyatiyu obosnovannyh reshenij po ekspertnym suzhdeniyam [The morphological approach to making reasonable decisions based on expert judgements]. *Vestnik TvGU. Seriya: Prikladnaya Matematika*, 2019, no. 2, pp. 42-56. https://doi.org/10.26456/vtpmk531
9. Kharchenko E. A. Algoritm morfologicheskogo metoda ekspertnyh ocenok dlya resheniya zadachi prognozirovaniya [Algorithm of the morphological method of expert estimates for solving the forecasting problem]. *Computer tools in education*, 2023, no. 2, pp. 5-20. https://doi.org/10.32603/2071-2340-2023-2-5-20.

**Anton D. Petrov**

Master’s Degree Student of the Department of Information Security, Mos-cow Polytechnic University (Moscow Poly, Russia, 107023, Moscow, B. Semenovskaya St. 38); Developer of the Security Analysis Direction, «BiZone» Limited Liability Company («BiZone» LLC, Russia, 105066, Moscow, Olkhovskaya St., Bld. 2, 4), e-mail: antonp2@yandex.ru, ORCID ID: 0009-0007-9546-8544.

**Elena A. Kharchenko**

Senior Lecturer of the Department of Infocognitive Technologies, Moscow Polytechnic University (Moscow Poly, Russia, 107023, Moscow, B. Semenovskaya St. 38), e-mail: elenakhaa@yandex.ru, ORCID ID: 0000-0002-5082-4564.