

Определение весов для модели оценки конфигурационного дрейфа

К. О. Моисеев[✉], А. О. Трофимова

Санкт-Петербургский государственный электротехнический университет
«ЛЭТИ» им. В. И. Ульянова (Ленина), Санкт-Петербург, Россия

✉ vihuhool@vk.com

Аннотация. Предлагается математическая модель для количественной оценки конфигурационного дрейфа в ИТ-инфраструктуре. Каждая проверка конфигурации описывается нормированной шкалой соответствия, позволяющей учитывать как бинарные, так и градуированные параметры. Проверки группируются по непересекающимся категориям, что обеспечивает интерпретируемость и возможность декомпозиции результатов. Модель вводит иерархическую систему весов, позволяющую агрегировать оценки от уровня отдельных проверок до узлов, ролей и кластера в целом. Представлены формальные свойства индекса соответствия и индекса дрейфа, включая ограниченность, аддитивность и устойчивость к изменениям шкал. Приводятся рекомендации по формированию атомарных проверок и корректных шкал, а также демонстрационный пример вычисления индекса для сервера с несколькими категориями параметров. Предложенный подход формирует строгую теоретическую основу для последующего построения систем мониторинга и управления конфигурационным дрейфом.

Ключевые слова: конфигурационный дрейф, ИТ-инфраструктура, нормированная метрика, индекс соответствия, агрегирование, иерархические веса, шкалы измерения, математическая модель

Для цитирования: Моисеев К. О., Трофимова А. О. Определение весов для модели оценки конфигурационного дрейфа // Изв. СПбГЭТУ «ЛЭТИ». 2026. Т. 19, № 5. С. 47–56. doi: 10.32603/2071-8985-2026-19-5-47-56.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Original article

Determining Weights for the Evaluation Model of Configurational Drift

K. O. Moiseev[✉], A. O. Trofimova

Saint Petersburg Electrotechnical University, Saint Petersburg, Russia

✉ vihuhool@vk.com

Abstract. This paper introduces a mathematical model for quantitatively assessing configuration drift in IT infrastructure. Each configuration check is represented by a normalized compliance scale, supporting both binary and graded parameters. Checks are grouped into non-overlapping categories to ensure interpretability and enable decomposition of results. The model defines a hierarchical system of weights that allows aggregation from individual checks to nodes, roles, and clusters. Formal properties of the compliance index and the drift index are established, including boundedness, additivity, and robustness to scale transformations. Guidelines are provided for designing atomic checks and consistent measurement scales, and a demonstration example illustrates the calculation of the drift index for a server with multiple parameter categories. The proposed approach provides a rigorous theoretical foundation for further development of monitoring and management systems for configuration drift.

Keywords: configuration drift, IT infrastructure, normalized metric, compliance index, aggregation, hierarchical weights, measurement scales, math model.

For citation: Moiseev K. O., Trofimova A. O. A Determining Weights for the Evaluation Model of Configurational Drift // LETI Transactions on Electrical Engineering & Computer Science. 2026. Vol. 19, no. 5. P. 47–56. doi: 10.32603/2071-8985-2026-19-5-47-56.

Conflict of interests. The authors declare no conflicts of interests.

Введение. Оценка конфигурационного дрейфа основана на агрегировании множества частных показателей, отражающих степень отклонения инфраструктуры, программных компонентов и политик от эталонного состояния.

В [1] авторами была представлена математическая основа модели, формализующая понятия дрейфа, индекса соответствия S и интегрального показателя отклонения $D = 1 - S$. Эта модель задает структуру категорий и проверок, однако вопрос назначения весов и порогов, обеспечивающих корректную оценку и интерпретируемость результатов, остается открытым.

Корректное определение весов имеет ключевое значение. Именно оно задает баланс между категориями, определяет приоритетность найденных несоответствий и напрямую влияет на чувствительность модели к различным видам изменений. Неправильно выбранные веса способны исказить оценку дрейфа: завязать значимость второстепенных аспектов или, наоборот, «приглушить» критичные. При этом даже небольшие сдвиги в весах могут существенно повлиять на агрегированный результат. Поэтому формирование весовых коэффициентов должно быть не интуитивным, а воспроизводимым и проверяемым процессом, основанным на прозрачных процедурах.

Для решения этой задачи используется метод анализа иерархий, который позволяет структурировать систему показателей и перевести экспертные представления о важности факторов в количественные соотношения. Метод опирается на разбиение цели на уровни – категории и проверки – и на процедуру последовательных парных сравнений, по результатам которых вычисляются относительные приоритеты. Такой подход обеспечивает согласованность экспертных решений, прослеживаемость рассуждений и возможность последующего анализа устойчивости полученных весов.

Применение метода анализа иерархий в контексте модели конфигурационного дрейфа позволяет:

- формализовать процесс экспертной калибровки весов категорий и проверок;
- обеспечить воспроизводимость и прозрачность решений при обновлениях модели;
- задать основу для валидации и анализа чувствительности при изменении исходных параметров.

В дальнейшем изложении рассматриваются основные этапы построения и калибровки весов на основе метода анализа иерархий – от построения структуры иерархии до проверки согласован-

ности экспертных оценок и формирования итоговых весовых коэффициентов, применимых в модели конфигурационного дрейфа.

Обзор существующих решений. Современные инженерные практики и исследования в области управления конфигурациями в ИТ-инфраструктурах сосредоточены преимущественно на автоматизации выявления и устранения отклонений между фактическим и эталонным состояниями. В парадигме «инфраструктура-как-код» (Infrastructure as Code, IaC) конфигурационный дрейф рассматривается как естественное следствие динамики изменений, а его устранение реализуется через периодические проверки и автоматическое приведение системы к эталону. Инструменты управления конфигурациями – Ansible, Puppet, Chef, SaltStack – обеспечивают высокий уровень автоматизации, однако отчеты таких систем, как правило, сводятся к бинарным выводам «соответствует/не соответствует». Это затрудняет ответ на вопросы о размере отклонения, вкладе отдельных компонент и динамике дрейфа во времени. Русскоязычные обзоры и прикладные публикации в основном описывают функциональность и сценарии применения этих инструментов, не предлагая формализованной количественной метрики состояния конфигураций [2].

В международных работах фокус смещен к раннему обнаружению и предупреждению о дрейфе в облачных и IaC-средах, включая анализ журналов IaC/облачных сервисов и применение методов искусственного интеллекта (ИИ) и машинного обучения (МО) для распознавания расхождений (например, на данных AWS Config) – т. е. решается задача «обнаружить/сигнализировать», но не «измерить единым сопоставимым показателем» [3], [4].

Дополняя этот пласт, систематические обзоры по IaC фиксируют зрелость практик и типичные проблемы качества артефактов IaC. Так, систематический обзор исследований IaC выявляет ключевые направления (фреймворки и инструменты, эмпирические исследования, тестирование), но не предлагает унифицированной числовой метрики конфигурационного состояния на уровне узлов и ролей [5], [6]. Систематический обзор практик («do's and don'ts») агрегирует нормы и антипаттерны из различных источников, что важно для профилактики дефектов инфраструктурного кода, однако остается в плоскости качественных рекомендаций [7]. Ряд эмпирических работ по IaC акцентирует внимание на рисках дефектов и опе-

рациональных отказов, возникающих из-за некорректных изменений конфигураций [8]. В сумме эти исследования закрывают вопросы, как писать и проверять IaC, но не дают интегральной количественной метрики дрейфа для сопоставления объектов и приоритизации исправлений.

В смежной области оценки устойчивости и деградации сложных систем следует отметить работу, посвященную анализу функциональной живучести структурно-сложных технических систем [9]. Авторы предлагают формализованный подход к оценке способности системы сохранять работоспособность при неблагоприятных воздействиях, переходя от бинарного понимания отказа к понятию сохраненной эффективности функционирования. Ключевым результатом работы становится выделение инвариантного ядра модели как вектора функциональной избыточности, который отражает количество допустимых работоспособных состояний системы при различном числе пораженных элементов и не зависит от конкретного сценария воздействия. Несмотря на то, что данный подход ориентирован на задачи живучести технических систем, а не на конфигурационный дрейф ИТ-инфраструктуры, он концептуально близок рассматриваемой модели в части иерархической декомпозиции, агрегирования частных состояний и интерпретируемой количественной оценки деградации системы.

Отдельная линия – безопасность облачно-нативных систем. Современные обзоры подчеркивают, что конфигурационный дрейф служит значимым фактором риска: разбалансировка параметров и политик по мере масштабирования и обновлений порождает уязвимости и несоответствие требованиям, что требует автоматизированного контроля и регулярной верификации. Тем не менее, и здесь фокус – на процессах и средствах обеспечения соответствия, а не на строгой нормированной метрике дрейфа, инвариантной к типам параметров и пригодной для декомпозиции по категориям.

Итого, существует множество разных подходов к обнаружению и устранению дрейфа (в том числе с ИИ-поддержкой), а также к качеству артефактов IaC и практикам их тестирования. Однако сохраняется методический разрыв: отсутствует общепринятая числовая, интерпретируемая и сопоставимая метрика дрейфа, которая корректно учитывает частичные соответствия, поддерживает иерархическое агрегирование от проверок к категориям/ролям/кластерам и позволяет выполнять разложение вклада по областям конфигурации для принятия управленческих решений.

На этом фоне особое значение приобретают методы, обеспечивающие оценку различных факторов и получение весов на основе экспертных суждений. В инженерных и управленческих дисциплинах для решения подобных задач широко используется метод анализа иерархий, предложенный Т. Саати [10]. Он позволяет переводить качественные экспертные оценки в количественные соотношения, контролировать согласованность суждений и строить воспроизводимые весовые модели. Позднейшие обзоры [11] отмечают, что метод анализа иерархий успешно применяется в технических и организационных системах для оценки рисков, выбора приоритетов и агрегирования метрик. В отечественной литературе разработаны подходы к комплексированию мнений экспертов и формированию групповых решений в рамках метода анализа иерархий [12], что важно при разнородности компетенций и необходимости консенсуса в инженерных задачах.

Применение метода анализа иерархий в задаче оценки конфигурационного дрейфа позволяет формализовать процесс калибровки весов категорий и проверок, обеспечить прослеживаемость решений и заложить основу для последующего анализа чувствительности и обновления модели при изменении технологического контекста.

Экспертная группа. Для определения весовых коэффициентов категорий параметров конфигурации была сформирована экспертная группа из шести специалистов, обладающих практическим опытом проектирования, эксплуатации и сопровождения серверных и инфраструктурных систем.

Состав экспертной группы обеспечивал покрытие ключевых аспектов управления ИТ-инфраструктурой, включая безопасность, надежность, производительность, наблюдаемость и архитектурное проектирование. Приведем краткое описание квалификации каждого эксперта:

Эксперт 1 – инженер по эксплуатации баз данных, стаж работы – 9 лет. Специализация – высоконагруженные СУБД (PostgreSQL, MySQL), отказоустойчивые кластеры, резервное копирование и восстановление после сбоев.

Эксперт 2 – инфраструктурный инженер, стаж работы – 11 лет. Специализация – проектирование и сопровождение серверной инфраструктуры, виртуализация, сетевые конфигурации, автоматизация управления инфраструктурой (Infrastructure as Code).

Эксперт 3 – администратор баз данных, стаж работы – 7 лет. Специализация – оптимизация производительности СУБД, анализ нагрузок, масштабирование и емкостное планирование.

Эксперт 4 – специалист по информационной безопасности, стаж работы – 10 лет. Специализация – «усиление» серверных систем, устранение уязвимостей, контроль доступа, обеспечение соответствия требованиям стандартов безопасности.

Эксперт 5 – инженер по надежности сервисов, стаж работы – 8 лет. Специализация – мониторинг и наблюдаемость, управление инцидентами, эксплуатация распределенных систем и сервисов.

Эксперт 6 – системный архитектор, стаж работы – 12 лет. Специализация – проектирование архитектур ИТ-систем.

Таким образом, экспертная группа объединяет практический опыт эксплуатации и архитектурного проектирования, что позволяет рассматривать задачу определения весов категорий параметров с различных профессиональных позиций.

Метод агрегации экспертных оценок. Для определения относительной важности категорий параметров конфигурации использовался метод анализа иерархий, основанный на попарных сравнениях критериев по шкале Саати.

Каждый эксперт формирует матрицу попарных сравнений критериев второго уровня. Индекс согласованности (ИС) для матрицы определяется по следующей формуле:

$$ИС = \frac{\lambda_{\max} - n}{n - 1},$$

где n – размер матрицы; λ_{\max} – значение, полученное суммированием произведений компонент локального вектора приоритетов и соответствующих им значений суммы j -го столбца матрицы попарных сравнений, где i для компоненты локального вектора приоритетов равняется j для столбца матрицы ПС.

Локальные приоритеты выставляются каждым экспертом в соответствии с вербально-числовой шкалой Саати, представленной в табл. 1.

Табл. 1. Вербально-числовая шкала Саати
 Tab. 1. Saaty's verbal-numerical scale

a_{ij}	Пояснения
1	Равная важность сравниваемых элементов иерархии
3	Умеренное превосходство i -го элемента иерархии над j -м
5	Существенное или сильное превосходство i -го элемента
7	Значительное превосходство i -го элемента
9	Очень значительное превосходство i -го элемента
2, 4, 6, 8	Промежуточные степени превосходства

Для получения итоговой матрицы попарных сравнений индивидуальные экспертные матрицы агрегировались методом геометрического среднего поэлементно.

Критерии оценки. В качестве критериев (категорий параметров конфигурации) использовался фиксированный набор из семи категорий. Каждая проверка конфигурации относится ровно к одной категории, категории считаются непересекающимися.

Описание категорий параметров конфигурации:

1. SEC (security) – безопасность. Безопасность доступа и узла – политики аутентификации и авторизации, механизмы разграничения прав, управление привилегиями, контроль уязвимостей, ограничения доступа к сервисам.

2. COMPL (compliance) – соответствие требованиям. Соответствие конфигурации внешним и внутренним нормативным требованиям и регламентам – международные и отраслевые стандарты (CIS, ISO, ГОСТ), корпоративные политики, требования регуляторов.

3. AVAIL (availability) – доступность и отказоустойчивость. Обеспечение доступности сервисов и устойчивости к отказам – проверки состояния, репликация, механизмы автоматического переключения, параметры восстановления.

4. PERF (performance) – производительность и емкость. Характеристики производительности и ресурсной емкости – задержки и пропускная способность, загрузка процессора, памяти и подсистемы ввода-вывода, метрики.

5. STACK – целостность программного стека. Корректность и согласованность программного стека – состав и версии пакетов, зависимости между компонентами, конфигурация окружений, соответствие эталонному стеку.

6. OBS (observability) – наблюдаемость и аудит. Средства мониторинга и контроля – сбор и полнота журналов событий, метрик и трассировок, аудит действий пользователей и процессов, корректность и непрерывность телеметрии.

7. NET (network) – сетевая конфигурация и связность. Параметры сетевой конфигурации и связности – сегментация и изоляция сетей, правила фильтрации трафика (ACL, firewall), открытые порты, настройки DNS и маршрутизации, использование сетевых криптографических протоколов (TLS).

Типы серверов (роли). Экспертная оценка весов категорий параметров выполнялась отдельно для различных типов серверов, что позволяет учитывать ролевую специфику эксплуатации и требований.

Использовался следующий набор ролей:

- DB (database server) – сервер баз данных;
- WEB (web server) – веб-сервер;
- APP (application server) – сервер приложений;
- OBS (observability server) – сервер мониторинга, логирования и метрик;
- INFRA (infrastructure server) – инфраструктурные сервисы (DNS, LDAP/AD, NTP, PKI, CI/CD...).

Выбранный набор ролей обеспечивает баланс между полнотой моделирования и трудоемкостью экспертной оценки, позволяя продемонстрировать ролевую специфику весовых коэффициентов без избыточного усложнения процедуры экспертизы.

Поиск весов категорий для различных типов серверов. Определение весовых коэффициентов категорий параметров конфигурации выполнялось с учетом ролевой специфики серверов. Для каждого типа серверов экспертами формировались отдельные матрицы попарных сравнений категорий параметров, отражающие относительную важность соответствующих аспектов эксплуатации и управления.

Каждый из шести экспертов независимо формировал матрицу попарных сравнений (м. п. с) категорий параметров конфигурации размером, используя девятибалльную шкалу Саати. Матрицы имели размер 7×7 . Для каждой матрицы вычислялись локальные векторы приоритетов категорий параметров перемножением элементов частных матриц попарного сравнения согласно

правилу, где элемент A_{ij} групповой м. п. с. рассчитывается по следующей формуле:

$$A_{ij} = n \sqrt[n]{\prod_{k=1}^n a_{ij}(k)},$$

где a_{ij} – элемент частной м. п. с для эксперта k . Дополнительно оценивалась согласованность экспертных суждений.

Также в промежуточных расчетах требуется вычислять:

- сумму по столбцу Σ ;
- значения локального вектора приоритетов по строкам (рассчитываются как частное от корня степени количества экспертов из произведения значений попарных сравнений по строке и суммы этих корней по всем строкам);
- произведение суммы по столбцу и соответствующего критерию значения вектора локальных приоритетов λ .

Для контроля согласованности также рассчитывается общая согласованность (ОС) как частное от значения индивидуальной согласованности (ИС) и значения по таблице случайной согласованности, в соответствии с количеством экспертов. Значение общей согласованности должно быть меньше 0.1.

Расчеты для разных типов серверов приведены в табл. 2–6.

Итоговые векторы приоритетов. Аналогичная процедура определения весов категорий параметров конфигурации была применена для всех типов серверов. Это позволило получить набор специфичных векторов весов, сопоставимых между собой за счет использования единой шкалы, фиксированного набора категорий и единой процедуры экспертной агрегации (табл. 7).

Табл. 2. Групповая матрица попарных сравнений для сервера БД
Tab. 2. Group pairwise comparison matrix for DB server

Критерий	SEC	COMPL	AVAIL	PERF	STACK	OBS	NET
SEC	1	6/1	5/6	3/2	6/1	3/1	7/4
COMPL	1/6	1	2/9	9/32	6/5	1/2	2/5
AVAIL	6/5	9/2	1	7/4	5/1	3/1	7/4
PERF	2/3	32/9	4/7	1	4/1	6/5	8/7
STACK	1/6	5/6	1/5	1/4	1	5/13	8/31
OBS	1/3	2/1	1/3	5/6	13/5	1	3/4
NET	4/7	5/2	4/7	7/8	31/8	4/3	1
Σ	4.17	20.37	3.74	6.47	23.43	10.45	7.04
λ	1.00	1.19	0.90	1.01	1.21	1.13	1.01
λ_{\max}	7.46						
ИС	0.08						
ОС	0.06						

Табл. 3. Групповая матрица попарных сравнений для сервера веб-сервера
 Tab. 3. Group pairwise comparison matrix for web-server

Критерий	SEC	COMPL	AVAIL	PERF	STACK	OBS	NET
SEC	1	5/1	3/2	5/3	9/2	7/3	1/1
COMPL	1/5	1	1/4	2/7	3/4	4/9	6/29
AVAIL	2/3	4/1	1	5/3	4/1	2/1	2/3
PERF	3/5	7/2	3/5	1	7/2	7/5	1/2
STACK	2/9	4/3	1/4	2/7	1	3/5	5/18
OBS	3/7	11/5	1/2	5/7	5/3	1	6/17
NET	1/1	29/6	3/2	2/1	18/5	17/6	1
Σ	4.10	21.87	5.63	7.63	19.07	10.58	4.01
λ	0.92	1.15	1.05	1.10	1.15	1.09	0.91
λ_{\max}	7.38						
ИС	0.06						
ОС	0.05						

Табл. 4. Групповая матрица попарных сравнений для сервера приложений
 Tab. 4. Group pairwise comparison matrix for app-server

Критерий	SEC	COMPL	AVAIL	PERF	STACK	OBS	NET
SEC	1/1	5/1	3/2	2/1	4/1	4/3	5/2
COMPL	1/5	1/1	7/19	5/14	5/8	4/11	3/8
AVAIL	2/3	19/7	1/1	8/5	3/1	7/5	2/1
PERF	1/2	14/5	5/8	1/1	9/4	5/8	6/5
STACK	1/4	8/5	1/3	4/9	1/1	7/15	9/16
OBS	3/4	11/4	5/7	8/5	15/7	1/1	5/4
NET	2/5	8/3	1/2	5/6	16/9	4/5	1/1
Σ	3.77	18.56	5.05	7.85	14.80	6.02	8.89
λ	0.93	1.18	0.95	1.06	1.17	0.99	1.10
λ_{\max}	7.38						
ИС	0.06						
ОС	0.05						

Табл. 5. Групповая матрица попарных сравнений для сервера мониторинга
 Tab. 5. Group pairwise comparison matrix for obs-server

Критерий	SEC	COMPL	AVAIL	PERF	STACK	OBS	NET
SEC	1	5/1	3/5	3/2	3/1	1/3	9/5
COMPL	1/5	1	7/29	3/10	9/16	1/6	3/11
AVAIL	5/3	29/7	1	2/1	7/2	1/2	8/5
PERF	2/3	10/3	1/2	1	9/4	7/16	9/8
STACK	1/3	16/9	2/7	4/9	1	2/7	1/2
OBS	3/1	6/1	2/1	16/7	7/2	1	5/2
NET	5/9	11/3	5/8	8/9	2/1	2/5	1
Σ	7.39	24.92	5.24	8.48	15.74	3.14	8.78
λ	1.14	1.21	1.03	1.06	1.13	0.88	1.09
λ_{\max}	7.54						
ИС	0.09						
ОС	0.07						

Табл. 6. Групповая матрица попарных сравнений для сервера инфраструктурных сервисов
 Tab. 6. Group pairwise comparison matrix for infra-server

Критерий	SEC	COMPL	AVAIL	PERF	STACK	OBS	NET
SEC	1	5/1	1/1	5/1	9/2	3/1	3/1
COMPL	1/5	1	5/14	3/1	8/3	2/1	2/1
AVAIL	1/1	14/5	1	5/1	4/1	3/1	5/2
PERF	1/5	1/3	1/5	1	2/1	1/2	5/8
STACK	2/9	3/8	1/4	1/2	1	7/16	1/2
OBS	1/3	1/2	1/3	2/1	16/7	1	1/1
NET	1/3	1/2	2/5	8/5	2/1	1/1	1
Σ	3.23	10.54	3.57	18.09	18.42	10.94	10.63
λ	0.93	1.35	0.92	1.24	1.06	1.09	1.07
λ_{\max}	7.66						
ИС	0.11						
ОС	0.08						

Табл. 7. Векторы приоритетов для разных типов серверов
 Tab. 7. Priority vectors for different types of servers

Критерий	Локальный вектор приоритетов				
	DB	WEB	APP	OBS	INFRA
Безопасность	0.24	0.23	0.24	0.16	0.29
Соответствие требованиям	0.06	0.05	0.06	0.05	0.13
Доступность и отказоустойчивость	0.24	0.19	0.19	0.20	0.26
Производительность	0.16	0.14	0.14	0.13	0.07
Целостность программного стека	0.05	0.06	0.08	0.07	0.06
Наблюдаемость и аудит	0.11	0.10	0.17	0.28	0.10
Сетевая конфигурация и связность	0.14	0.23	0.12	0.12	0.10

Для сервера баз данных (DB) наибольшую важность имеют безопасность и доступность, поскольку он хранит критичные данные и должен обеспечивать их защиту и непрерывный доступ без потерь. Веб-серверу (WEB) в первую очередь требуются безопасность и корректная сетевая конфигурация, так как он выступает внешней точкой входа и наиболее подвержен сетевым атакам и перегрузкам. Для сервера приложений (APP) приоритетны безопасность и наблюдаемость, поскольку он служит центральным элементом обработки запросов и связующим звеном между пользовательским уровнем и хранилищем данных, что требует контроля как корректности выполнения сервисов, так и возможных инцидентов. В случае сервера мониторинга (OBS) приоритет отдан наблюдаемости и доступности, так как его основная функция заключается в непрерывном сборе и анализе информации о состоянии всей инфраструктуры. Для сервера инфраструктурных сервисов (INFRA) доминируют безопасность и отказоустойчивость, поскольку сбои или компрометация базовых сервисов оказывают системное влияние на работоспособность всех остальных компонентов ИТ-среды.

Использование векторов приоритетов в качестве весовых коэффициентов. Полученные в результате экспертной калибровки векторы приоритетов категорий параметров конфигурации используются в модели оценки конфигурационного дрейфа в качестве весовых коэффициентов при агрегировании частных показателей.

Корректность такого использования обеспечивается следующими свойствами полученных векторов:

- все компоненты векторов неотрицательны;
- сумма компонентов каждого вектора равна единице;
- веса получены в рамках единой шкалы и единой процедуры экспертной агрегации;
- согласованность экспертных суждений подтверждена значениями отношения согласованности Саати, не превышающими порог $ОС < 0.1$.

Тем самым векторы приоритетов удовлетворяют требованиям нормированности и интерпретируемости и могут быть непосредственно подставлены в формулы агрегирования индекса соответствия и индекса дрейфа, введенные в первой части исследования.

Рисунок наглядно демонстрирует распределение весовых коэффициентов категорий параметров конфигурации для различных типов серверов в виде тепловой карты, что позволяет визуально сопоставить их относительную значимость. Из изображения четко видно, что приоритеты категорий существенно варьируются в зависимости от роли сервера. Так, для серверов баз данных наибольший вклад в интегральную оценку вносят категории безопасности и доступности, тогда как для веб-серверов и инфраструктурных сервисов возрастает значимость сетевой конфигурации и параметров периметра. Для серверов мониторинга и наблюдаемости доминирующее положение занимают параметры категории OBS, что отражает их функциональное назначение.

Таким образом, полученные векторы приоритетов не служат универсальными «абстрактными» весами, а формируют набор согласованных ролевых профилей, обеспечивающих сопоставимость оценок при сохранении семантики конкретного типа сервера.

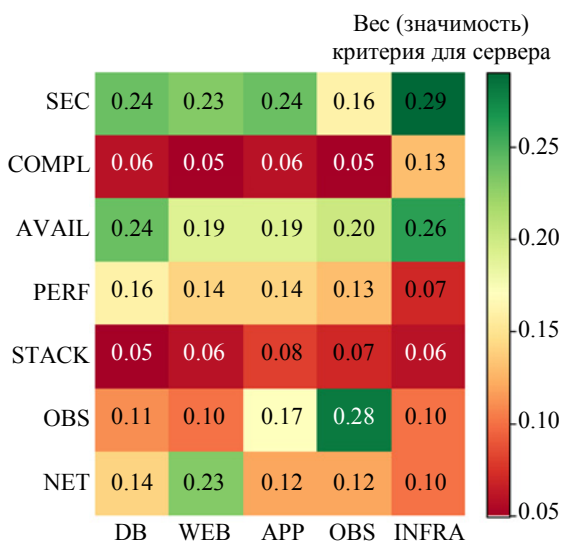


Рис. Тепловая карта для сравнения значимости категорий

Fig. Heatmap for competitive category comparison

Заключение. В статье предложен формальный и воспроизводимый подход к оценке конфигурационного соответствия и дрейфа серверных систем, основанный на использовании единой нормированной метрики и специфичных весовых коэффициентов категорий параметров. В отличие от традиционных чек-листов и разрозненных наборов проверок, предложенный подход обеспечивает сопоставимость результатов для разнородных серверов и поддерживает иерархическое разложение интегральной оценки по категориям и уровням инфраструктуры.

Разработана и обоснована методика определения весов категорий параметров на основе групповой экспертной оценки с применением метода анализа иерархий. Использование геометрического среднего для агрегации экспертных суждений и контроль согласованности по показателю Саати позволили получить устойчивые, нормированные и интерпретируемые векторы весов, пригодные для прямого использования в формальных моделях.

Показано, что распределение весов существенно зависит от функциональной роли сервера. Для серверов баз данных и инфраструктурных сервисов наибольшую значимость имеют параметры безопасности и доступности, для серверов мониторинга – параметры наблюдаемости и аудита, для веб-серверов – параметры сетевой конфигурации, тогда как серверы приложений характеризуются более сбалансированным профилем приоритетов. Эти результаты подтверждают принципиальную несостоятельность универсальных весов и обосновывают необходимость ролево-специфичных профилей.

Практическая значимость исследований заключается в возможности интеграции предложенной методики в существующие процессы мониторинга и аудита конфигураций без их радикальной перестройки. Полученные весовые коэффициенты могут использоваться для агрегирования показателей, приоритизации корректирующих действий и поддержки принятия решений в эксплуатации ИТ-инфраструктуры.

Перспективы дальнейшего развития представленного подхода связаны с расширением модели за счет учета временной динамики конфигурационного дрейфа и анализа устойчивости весовых коэффициентов при изменении эксплуатационного контекста. Представляется целесообразным исследовать возможности автоматизированной корректировки весов на основе накопленных данных мониторинга и инцидентов, что позволит дополнить экспертную калибровку эмпирическими механизмами адаптации. Дополнительное направление развития – это расширение набора серверных ролей и категорий параметров, а также разработка программной системы, реализующей предложенную модель и обеспечивающей автоматизированный сбор конфигурационных данных, расчет индексов соответствия и дрейфа, визуализацию результатов и поддержку принятия решений при эксплуатации реальных ИТ-инфраструктур.

Список литературы

1. Моисеев К. О., Трофимова А. О. Математическая модель для измерения конфигурационного дрейфа в ИТ-инфраструктуре // Изв. СПбГЭТУ «ЛЭТИ». 2026. Т. 19, № 1. С. 52–59. doi: 10.32603/2071-8985-2026-19-1-52-59.
2. Костромин Р. О. Сравнительный обзор средств управления конфигурациями ресурсов вычислительной среды функционирования цифровых двойников // Информационные и математические технологии в науке и управлении. 2021. № 1(21). С. 131–145. doi: 10.38028/ESI.2021.21.1.011.
3. Thiyagarajan G., Bist V., Nayak P. AI-driven configuration drift detection in cloud environments // Intern. J. of Communication Networks and Inform. Security (IJCNIS). 2024. Vol. 16, no. 5. P. 721–743. URL: <https://www.ijcnis.org/index.php/ijcnis/article/view/7898> (дата обращения: 24.09.2025).
4. Pohjola O. Mitigating configuration drift in infrastructure-as-code systems // Master's Thesis, Aalto University. 2025. URL: <https://aaltodoc.aalto.fi/bitstreams/6a2f20a2-d7cb-4037-9453-22d28fb0f10a/download> (дата обращения: 24.09.2025).
5. Security in cloud-native services: A survey / T. Theodoropoulos, L. Rozsa, C. Benzaid, P. Gray, E. Marin, A. Makris, L. Cordeiro, F. Diego, P. Sorokin, M. Di Girolamo, P. Barone, T. Taleb, C. Cérin // J. of Cybersecurity and Privacy. 2023. Vol. 4, no. 3. P. 758–793. doi: 10.3390/jcp4030034.
6. Rahman A., Mahdavi-Hezaveh R., Williams L. A systematic mapping study of infrastructure as code research // Inform. and Software Technol. 2019. Vol. 108. P. 65–77. doi: 10.1016/j.infsof.2018.12.004.
7. The do's and don'ts of infrastructure code: A systematic gray literature review / I. Kumara, M. Garriga, A. U. Romeu, D. Di Nucci, D. A. Tamburri, W-J. van den Heuvel, F. Palomba // Inform. and Software Technol. 2021. Vol. 137, no. 9. P. 106593. doi: 10.1016/j.infsof.2021.106593.
8. Rahman A., Farhana E., Williams L. The 'as code' activities: Development anti-patterns for infrastructure as code // Empirical Software Engin. 2020. Vol. 25, no. 5. P. 3430–3467. doi: 10.1007/s10664-020-09841-8.
9. Черкесов Г. Н., Недосекин А. О., Виноградов В. В. Анализ функциональной живучести структурно-сложных технических систем // Надежность. 2018. Т. 18, № 2. С. 17–24. doi: 10.21683/1729-2646-2018-18-2-17-24.
10. Саати Т. Л. Принятие решений. Метод анализа иерархий / пер. с англ. Р. Г. Вачнадзе. М.: Радио и связь, 1993. 320 с.
11. Ishizaka A., Labib A. Review of the main developments in the analytic hierarchy process // Expert Syst. with Appl. 2011. Vol. 38, no. 11. P. 14336–14345. doi: 10.1016/j.eswa.2011.04.143.
12. Падерно П. И. Комплексование мнений экспертов при экспертной оценке с применением метода анализа иерархий // Человеческий фактор. Проблемы психологии и эргономики. 2005. № 3/1. С. 60–61.

Информация об авторах

Моисеев Кирилл Олегович – магистрант кафедры информационных систем СПбГЭТУ «ЛЭТИ».

E-mail: vihuhool@vk.com

<https://orcid.org/0000-0001-8788-6470>

Трофимова Анастасия Олеговна – магистрант кафедры информационных систем СПбГЭТУ «ЛЭТИ».

E-mail: anastasiia_trofimova02@mail.ru

References

1. Moiseev K. O., Trofimova A. O. Matematicheskaja model dlja izmerenija konfiguracionnogo drejfa v IT-infrastrukture // Izv. SPbGETU «LETI». 2026. T. 19, № 1. S. 52–59. doi: 10.32603/2071-8985-2026-19-1-52-59. (In Russ.).
2. Kostromin R. O. Sravnitelnyj obzor sredstv upravlenija konfiguracijami resursov vychislitelnoj sredy funkcionirovanija cifrovyh dvojnikov // Informacionnye i matematicheskie tehnologii v nauke i upravlenii. 2021. No. 1(21). S. 131–145. doi: 10.38028/ESI.2021.21.1.011. (In Russ.).
3. Thiyagarajan G., Bist V., Nayak P. AI-driven configuration drift detection in cloud environments // Intern. J. of Communication Networks and Inform. Security (IJCNIS). 2024. Vol. 16, no. 5. P. 721–743. URL: <https://www.ijcnis.org/index.php/ijcnis/article/view/7898> (data obrashhenija: 24.09.2025).
4. Pohjola O. Mitigating configuration drift in infrastructure-as-code systems // Master's Thesis. Aalto University. 2025. URL: <https://aaltodoc.aalto.fi/bitstreams/6a2f20a2-d7cb-4037-9453-22d28fb0f10a/download> (data obrashhenija: 24.09.2025).
5. Security in Cloud-Native Services: A Survey / T. Theodoropoulos, L. Rozsa, C. Benzaid, P. Gray, E. Marin, A. Makris, L. Cordeiro, F. Diego, P. Sorokin, M. Di Girolamo, P. Barone, T. Taleb, C. Cérin // J. of Cybersecurity and Privacy. 2023. Vol. 4, no. 3. P. 758–793. doi: 10.3390/jcp4030034.
6. Rahman A., Mahdavi-Hezaveh R., Williams L. A systematic mapping study of infrastructure as code research // Inform. and Software Technol. 2019. Vol. 108. P. 65–77. doi: 10.1016/j.infsof.2018.12.004.

7. The do's and don'ts of infrastructure code: A systematic gray literature review / I. Kumara, M. Garriga, A. U. Romeu, D. Di Nucci, D. A. Tamburri, W.-J. van den Heuvel, F. Palomba // Inform. and Software Technol. 2021. Vol. 137, no. 9. P. 106593. doi: 10.1016/j.infsof.2021.106593.

8. Rahman A., Farhana E., Williams L. The 'as code' activities: development anti-patterns for infrastructure as code // Empirical Software Engin. 2020. Vol. 25, no. 5. P. 3430–3467. doi: 10.1007/s10664-020-09841-8.

9. Cherkosov G. N., Nedosekin A. O., Vinogradov V. V. Analiz funkcionalnoj zhivuchesti strukturno-slozhnyh tehnicheskikh sistem // Nadezhnost. 2018. T. 18, № 2.

S. 17–24. doi: 10.21683/1729-2646-2018-18-2-17-24. (In Russ.).

10. Saati T. L. Prinjatje reshenij. Metod analiza ierarhij / per. s angl. R. G. Vachnadze. M.: Radio i svjaz, 1993. 320 s. (In Russ.).

11. Ishizaka A., Labib A. Review of the main developments in the analytic hierarchy process // Expert Syst. with Appl. 2011. Vol. 38, no. 11. P. 14336–14345. doi: 10.1016/j.eswa.2011.04.143.

12. Paderno P. I. Kompleksirovanie mnenij ekspertov pri ekspertnoj ocenke s primeneniem metoda analiza ierarhij // Chelovecheskij faktor. Problemy psihologii i ergonomiki. 2005. № 3/1. S. 60–61. (In Russ.).

Information about the authors

Kirill O. Moiseev – Master's degree student, Department of Information Systems, Saint Petersburg Electrotechnical University.

E-mail: viuhool@vk.com

Anastasiia O. Trofimova – Master's degree student, Department of Information Systems, Saint Petersburg Electrotechnical University.

E-mail: anastasiia_trofimova02@mail.ru

Статья поступила в редакцию 20.02.2026; принята к публикации после рецензирования 31.03.2026; опубликована онлайн 25.05.2026.

Submitted 20.02.2026; accepted 31.03.2026; published online 25.05.2026.
