УДК 004.94(07)

Алгоритм построения модели корреляции отказов телекоммуникационных сетей на основе леммы Йонеды и алгоритма ECLAT

Васильев Н. В., Буравлев А. С., Будко Н. П.

Постановка задачи*: современные телекоммуникационные сети формируют огромные объемы событийных данных, среди которых выделяются отказы и предупреждения. Эффективная корреляция таких событий является ключевым условием для своевременной диагностики и устранения неисправностей. Актуальность исследования обусловлена необходимостью разработки методов, позволяющих автоматически выявлять причинно-следственные зависимости между событиями в условиях высокой сложности, наложения шумовых данных и параллельных процессов в распределенной инфраструктуре.* ***Целью работы*** *является построение формальной модели и алгоритма автоматической реконструкции графов зависимостей событий отказов сетевых элементов на основе анализа журналов мониторинга. В работе представлено описание формализма графов зависимостей событий, описан предложенный алгоритм построения графов на основе суффиксных деревьев, а также приведено краткое описание и результаты вычислительных экспериментов по влиянию шумов на качество реконструкции модели реализованным прототипом. Предложенный алгоритм включает этапы построения граней событий, формирования матрицы отношений между гранями, а также фильтрации и построения результирующей модели. Для построения граней событий используются ECLAT-суффиксные деревья, применяемые к транзакциям журнала событий системы мониторинга. Транзакции выделяются на основе эвристики симметрии события отказа и события восстановления работоспособности. На следующем этапе производится анализ условных вероятностей включения граней друг в друга с учетом возможного параллелизма событий и фильтрацией маловероятных отношений. На заключительном этапе сформированная модель отношений между гранями преобразуется в граф зависимостей событий.* ***Используемые методы:*** *методы вычислительной теории графов; методы анализа данных; теория вероятностей и математическая статистика; системный анализ; теория моделирования систем; теория множеств и теория категорий.* ***Научная новизна*** *предложенного подхода заключается в разработанной концепции зависимости событий на основе включения наименьших верхних граней элементов предпорядка, что обеспечивает возможность формирование графов зависимостей событий в условиях зашумленных взаимными наложениями цепочек событий.* ***Результат*** *исследования заключается в возможности автоматизации анализа событийных журналов телекоммуникационных сетей, снижении нагрузки на обслуживающий персонал и ускорении поиска первопричин отказов. Экспериментальные результаты показывают, что предложенный метод эффективен при низком и умеренном уровне наложения событий, обеспечивая высокую точность восстановления зависимостей. Таким образом, работа формирует основу для создания интеллектуальных систем мониторинга, способных выявлять скрытые закономерности и поддерживать надежность функционирования распределенных информационно-телекоммуникационных систем.*

Ключевые слова: *ассоциативные правила, графы зависимостей событий, извлечение процессов, корреляция и фильтрация событий, лемма Йонеды, частично упорядоченные множества.*

Библиографическая ссылка на статью:
Васильев Н. В., Буравлев А. С., Будко Н. П. Алгоритм построения модели корреляции отказов телекоммуникационных сетей на основе леммы Йонеды и алгоритма ECLAT // Системы управления, связи и безопасности. 2025. № 4. С. 27-46. DOI: 10.24412/2410-9916-2025-4-027-046

Reference for citation:
Vasiliev N. V., Buravlev A. S., Budko N. P. Algorithm for constructing a correlation model for failures of telecommunication networks based on Yoneda's lemma and the ECLAT algorithm. *Systems of Control, Communication and Security*, 2025, no. 4, pp. 27-46 (in Russian). DOI: 10.24412/2410-9916-2025-4-027-046

Введение

В контексте телекоммуникационных сетей (ТКС) под отказом понимается выход из строя какого-либо компонента (сетевого элемента, сети, соединения, сервиса), приводящий к некорректной работе системы. Отказ, как правило, влечет за собой возникновение ряда косвенных событий, на основании которых система мониторинга должна вынести заключение о состоянии телекоммуникационной системы посредством модулей корреляции и фильтрации, отсеивающих второстепенные и косвенные события о неисправностях. Работа этих модулей основана на установлении причинно-следственных связей между множеством событий.

Для решения задачи корреляции и фильтрации событий было предложено несколько подходов, краткое описание, их преимущества и недостатки приведены в таблице 1.

Таблица 1 – Основные подходы к корреляции событий отказов ТКС

Подход	Пример работ / систем	Используемая модель	Достоинства	Недостатки
Системы на основе правил (Rule-based)	[1] / IBM Netcool	Продукционные правила: ЕСЛИ *X* и *Y*, то *Z*	Простота реализации, понятность	Потеря актуальности при изменении сети, требует ручного сопровождения
Базы знаний и экспертные системы	[2] / HP OView	Базы знаний, деревья причин отказов, паттерны сбоев	Учитывают доменные знания, интерпретируемость	Трудоёмкое пополнение знаний, плохо масштабируются
Графы зависимостей событий	[3, 5]	Графы зависимостей событий, байесовские сети	Быстрое выявление базового множества отказов, структурный подход	Построение и обновление моделей сложно, требуется топология
Автоматные и формальные модели	[4]	Конечные автоматы, регулярные выражения, алгебра событий	Хорошо для временных последовательностей, формализуемо	Сложно покрыть все реальные сценарии, экспоненциальный рост состояний
Data Mining и машинное обучение	[6] / IBM Proactive Net Mgmt, Splunk ITSI, Moogsoft	Трассировка журналов, кластеризация, байесовские сети, нейросети	Автоматическое выявление паттернов, масштабируемость	Требуют данных для обучения, интерпретация часто сложна
Гибридные системы	[7] / Moogsoft	Многоуровневые фильтры + графы + ML	Снижение шума, устойчивость, адаптивность	Сложность реализации и сопровождения

Практически все представленные в таблице модели (правила, базы знаний, графы) требуют постоянного обновления при изменении топологии, сервисов и оборудования, что говорит о высокой сложности их сопровождения. Каждый метод имеет применимость в определенных сценариях:

– правила – для простых случаев;
– графы – для детерминированных зависимостей;
– машинное обучение – при наличии больших датасетов.

Нет единого подхода, который решал бы все задачи. При росте числа устройств и событий модели часто становятся громоздкими (взрыв количества правил, состояний, возможных комбинаций). В то время как классические методы (правила, графы) понятны, но неустойчивы при изменениях структуры сети. Современные гибридные подходы устойчивее, но их выводы труднее объяснить. Совокупность общих недостатков существующих подходов – высокая стоимость сопровождения вследствие зависимости от ручного обновления знаний, слабая масштабируемость. Рациональной стратегией является разработка алгоритма, в котором на стадии обучения применяются методы машинного обучения для автоматического выявления взаимного влияния отказов, а на стадии эксплуатации используется сформированная графовая модель зависимостей отказов, которая обеспечивает быстрый поиск первопричины и наглядную интерпретацию результатов. Такой подход сочетает адаптивность машинного обучения (самообновляемая база знаний без участия операторов) и структурность графов (понятность и скорость корреляции), что снижает нагрузку на администраторов и повышает устойчивость системы к изменениям в сети. Указанные соображения определяют цели и задачи настоящей работы.

Основные обозначения и терминологический аппарат

В работе вводятся следующие условные обозначения, показанные в таблице 2.

Таблица 2 – Основные обозначения

Обозначение	Физический смысл обозначения
e	Событие (например: отказ сетевого элемента, восстановление)
t_e	Время возникновения события
o_e	Объект, на котором возникло событие (сетевой элемент, соединение, сеть)
a_e	Действие, о котором сигнализирует событие (переход в неисправное состояние, восстановление, критическая ситуация и пр.)
$a \to b$	Возникновение отказа a приводит к отказу b (отношение причинности)
$a \parallel b$	Отказ a может происходить одновременно с отказом b вследствие наличия общего события-причины
$e \Rightarrow \{e_1, e_2, .., e_k\}$	Событие e коррелирует с множеством событий e_1, e_2, …, e_k
$\Im = \langle E, \to \rangle$	Структура событий (causal event structure) $\Im = \langle E, \to \rangle$, где E – множество событий, $\to \subseteq E \times E$ – причинно-следственное бинарное отношение, задающее слабый частичный порядок на множестве событий.
$\prec$	Транзитивное замыкание отношения $\to$: $e_a \prec e_b \Leftrightarrow \exists e_1, e_2 ... e_n \in E : e_a \to e_1 \to e_2 \to ... \to e_n \to e_b$

DOI: 10.24412/2410-9916-2025-4-027-046
URL: https://sccs.intelgr.com/archive/2025-04/02-Vasiliev.pdf

Обозначение	Физический смысл обозначения
StepEff(*e*)	Шаговый эффект события *e*
MinCause (*S*)	Минимальное множество причин множества событий *S*
$(X, \leq_X)$	Предпорядок на множестве *X*
$\uparrow p$	Верхняя грань элемента *p* предпорядка $(P, \leq_P)$: $\uparrow p = \{p' \in P \mid p \leq p'\}, p, p' \in P$
$U \subseteq P$	Верхняя грань предпорядка $(P, \leq_P)$ – подмножество $U \subseteq P$ такое, что если $p \in U, p \leq_P q \Rightarrow q \in U$
$U(P)$	Множество верхних граней предпорядка *P*
E	Множество событий отказов
L	Входной журнал
{*Trans*}	Множество транзакций события (участков от отказа до восстановления)
$\{F_e\}$	Частоты граней события *e*
$Edge_e$	Множество верхних граней события *e*
{*Edges*}	Множество верхних граней (в контексте вычисления для события)
{*E*,{*Edges*}}	Отображение (словарь) «событие» – «множество его граней»
$\#(\uparrow e)$	Число транзакций грани события *e* в журнале
$Supp(Edge_e)$	Поддержка грани *Edge* события *e* (частота заданного набора последующих событий)
$Conf(e_1, e_2)$	Максимальная достоверность (условная вероятность) включения грани $Edge_{e2}$ в грань $Edge_{e1}$
$Conf(\uparrow a \subset \uparrow b)$	Достоверность (условная вероятность) включения грани события *a* в грань события *b*
CMATR	Матрица достоверности зависимости событий
$Lift(\uparrow a \subset \uparrow b)$	Условие зависимости фактов наличия в транзакции событий *a* и *b*
$Conv(\uparrow a \subset \uparrow b)$	Убедительность включения грани события *a* в грань события *b*
support	Порог фильтрации граней событий журнала по поддержке
delcon	Порог фильтрации взаимно включенных граней событий журнала
α_1	Точность реконструкции графа зависимости событий
α_2	Избыточность графа зависимости событий

Графы зависимости событий

Событием с точки зрения мониторинга будем понимать всякое изменение состояния объекта (сетевого элемента), происходящее в определенный (дискретный) момент времени.

Введем базовые соглашения об атрибутах событий. Каждое событие *e* характеризуется тройкой атрибутов:

– время (t_e);
– объект (o_e);
– действие (a_e).

Объект и действие, взятые вместе, составляют идентификатор события. Далее, рассуждая о событиях, мы будем подразумевать именно их идентификаторы.

Среди потока событий, поступающего от элементов информационной инфраструктуры, по типу действия обычно выделяют:

– отказы (fault) – события, сигнализирующие о ненормальном состоянии сетевого элемента, т. е. таком состоянии, при котором реальное поведение элемента отличается от ожидаемого;
– предупреждения (alarms) – события, сигнализирующие о восстановлении состояния сетевого элемента, а также о пограничных состояниях (с возможными градациями: «незначительное» (minor), «существенное» (major), «критическое» (critical)).

Отказы в телекоммуникационной сети демонстрируют «*поведенческую симметрию*» в контексте журналов событий. В большинстве случаев, после регистрации отказа в журнале, спустя некоторое время можно наблюдать событие восстановления. Исключение составляют случаи необратимого разрушения или модификации телекоммуникационной системы, но они редки по отношению к первому типу событий. Это наблюдение используется в работе для выделения участков журналов, подлежащих анализу. В работе такие участки будем называть *транзакциями*.

Проблема моделирования связей между событиями в распределенной инфраструктуре может быть описана с помощью *частично упорядоченных множеств*. Действительно, наличие явных правил, таких как «возникновение отказа *a* приводит к отказу *b*», при отсутствии циклических отношений (любой длины) логично описать с помощью *отношений порядка* $a \rightarrow b$, представлением которых в совокупности может служить ациклический ориентированный граф. Для простоты далее будем называть такое отношение «*отношением причинности*». Введем несколько базовых определений и понятий [5]. Будем говорить, что событие *e коррелирует* с множеством событий $e_1, e_2, \ldots, e_k$, если $e_1, e_2, \ldots, e_k$ входят в отношение причинности друг с другом и с событием *e* и таким образом определяют *шаблон (паттерн) поведения* системы при появлении события *e*.

Будем обозначать этот факт $e \Rightarrow \{e_1, e_2, .., e_k\}$.

Структура событий (*causal event structure*) задается парой $\Im = \langle E, \rightarrow \rangle$, где E – множество событий, а $\rightarrow \subseteq E \times E$ причинно-следственное бинарное отношение, задающее слабый частичный порядок на множестве событий.

Т. е. данное отношение обладает следующими свойствами. Для всяких событий $e_a, e_b, e_c \in E$:

– $\dfrac{e_a \rightarrow e_b \ , \ e_b \rightarrow e_c}{e_a \rightarrow e_c}$ *транзитивность*;
– $e_a \not\rightarrow e_a$ *антисимметричность*.

Интуитивный смысл отношения $e_1 \rightarrow e_2$ – событие e_1 является причиной e_2 (возможно причиной нескольких событий).

Транзитивное замыкание отношения $\rightarrow$ обозначим $\prec$, именно

$$e_a \prec e_b \Leftrightarrow \exists e_1, e_2 ... e_n \in E : e_a \rightarrow e_1 \rightarrow e_2 \rightarrow ... \rightarrow e_n \rightarrow e_b$$

Наглядной формой представления структуры событий является событийный граф. Основным свойством данного графа является ацикличность, которая следует из транзитивности и антисимметричности структуры событий. В вершинах данного графа располагаются события, а ребра выражают причинно-следственное отношение. В качестве примера рассмотрим граф на рис. 1. Из приведенного рисунка следует, что $8 \Rightarrow \{2, 3, 6\}$, т. к. появление события 8 приведет к появлению 2, 3, 6. Также можем заключить, что $8 \Rightarrow \{1, 2, 3, 6\}$, что следует из транзитивности отношения причинности, так как появление события 2 приведет к появлению события 1. С другой стороны, появление событий {3, 6} еще не говорит о наличии события 8, так как для появления последнего необходимо также наличие события 2. По транзитивности же получаем, что $11 \Rightarrow \{4, 5, 10\}$.

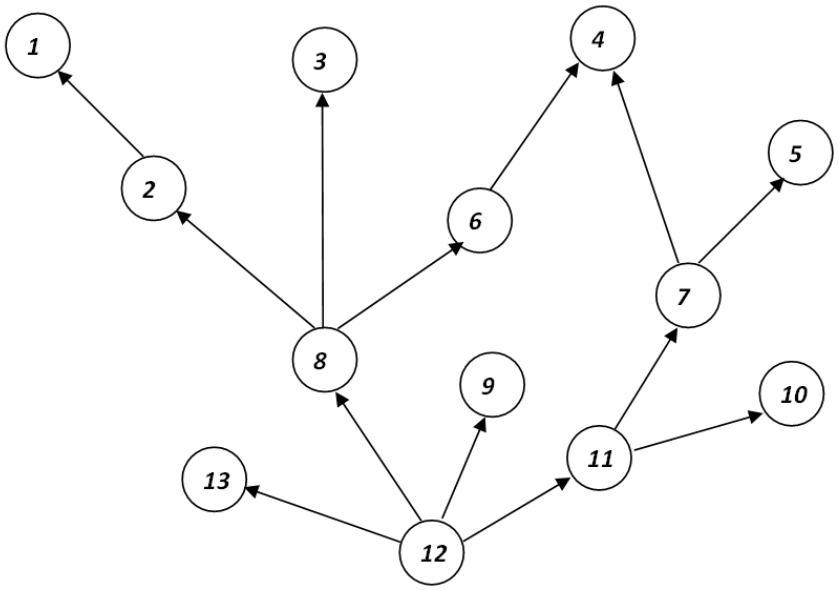

Рис. 1. Пример событийного графа

Пусть задана структура $\Im = \langle E, \to \rangle$. *Шаговым эффектом события* $e \in E$ *StepEff*(e) назовем множество событий $S = StepEff(e) = \{e' | e \to e'\}$

Расширим отношение $\Rightarrow$ до множеств. А именно $S \Rightarrow S'$, если

$$\forall e \in S \ \exists e' \in S' : e \prec e'$$

$$!\exists e \in S \ \forall e' \in S' : e' \prec e$$

Для формализации семантики оператора $\Rightarrow$ вводится следующие правила вывода [5]:

– (Effects) $\{e\} \Rightarrow StepEff(e)$ *if* $StepEff(e) \neq \varnothing$;

– (Reflexivity) $S \Rightarrow S$;

– (Transitivity) $\dfrac{S_1 \Rightarrow S_2 \quad S_2 \Rightarrow S_3}{S_1 \Rightarrow S_3}$;

– (Union) $\dfrac{S_1 \Rightarrow T_1 \quad S_2 \Rightarrow T_2}{S_1 \cup S_2 \Rightarrow T_1 \cup T_2}$.

Доказано, что в рамках описанной семантики, система правил вывода полна и непротиворечива. Указанная схема может быть использована для практической реализации модуля корреляции и фильтрации событий.

Например, для графа на рис. 1 можно показать, что $8 \Rightarrow \{1, 2, 3, 6\}$ исходя из $8 \Rightarrow \{2, 3, 6\}$ следующим образом:

1) $\{2\} \Rightarrow \{1\}$ (Effect)

2) $\{3, 6\} \Rightarrow \{3, 6\}$ (Reflexivity)

3) $\{2, 3, 6\} \Rightarrow \{1, 3, 6\}$ (Union)

4) $8 \Rightarrow \{2, 3, 6\}$, $\{2, 3, 6\} \Rightarrow \{1, 3, 6\}$, следовательно $8 \Rightarrow \{1, 2, 3, 6\}$ (Transitivity)

Следует заметить, что оператор $\Rightarrow$ не является простым расширением оператора $\rightarrow$ до множеств. В частности, согласно системе правил вывода оператор $\Rightarrow$ рефлексивен, в то время как $\rightarrow$ нет.

Пусть задана структура $\Im = \langle E, \rightarrow \rangle$. *Рангом события e* назовем величину:

$$Rank(e) = \begin{cases} 0 & \text{, если } StepEff(\text{e}) = \varnothing; \\ \text{Max}\left(\left\{Rank(e') \mid e' \in StepEff\ (e)\right\}\right) + 1 & \text{, в противном случае.} \end{cases}$$

Наглядно, ранг элемента равен высоте максимального поддерева элемента +1.

Ранг множества событий $S \subseteq E$ равен

$$Rank(S) = Max(\{Rank(e) \mid e \in S\}).$$

Обоснованность приведенной схемы корреляции и фильтрации событий обеспечивается следующей теоремой [5], которая гласит, что при заданной структуре $\Im = \langle E, \rightarrow \rangle$ и $S \subseteq E$ – подмножестве априорных событий, полученных из внешней среды, минимальное множество причин событий S $MinCause$ (S) должно обладать следующими свойствами:

1) *Корректность*: $MinCause(S) \Rightarrow S$;

2) *Оптимальность*: Для любого $S' : S' \Rightarrow S$ имеем

$$Rank(S') \leq Rank(MinCause(S)).$$

Преимущество использования событийных графов заключается в упрощении процесса логического вывода первопричин отказов. Анализ начинается с фиксации события, свидетельствующего о возникшей неисправности. Далее, посредством последовательного рекурсивного обхода графа в направлении, противоположном причинно-следственной связи, осуществляется идентификация событий, непосредственно предшествующих возникновению рассматриваемой неисправности. Данная процедура повторяется до тех пор, пока не будет обнаружено событие, не имеющее предшествующих причин, либо до достижения заранее определенной глубины поиска.

Простота структуры графа и четкость логики вывода обеспечивают возможность эффективного использования данного подхода, как в ручном режиме,

так и при создании автоматизированных систем мониторинга и управления. При этом следует учитывать ограничения, связанные с потенциальным наложением граней событий и наличием шумовых данных, которые могут повлиять на точность и скорость вывода.

В следующем разделе будет представлен метод автоматического построения графов зависимостей событий, основанный на алгоритмах построения ассоциативных правил и некоторых свойствах частично упорядоченных множеств. В последующем разделе проводится экспериментальная проверка устойчивости алгоритма к наложению независимых события выхода из строя сетевых элементов.

Теоретическое обоснование использования ассоциативных правил на журналах событий

Приведенная концептуальная модель корреляции и фильтрации событий может быть рассмотрена с точки зрения результатов, полученных для предпорядков и частично-упорядоченных множеств [8, 9].

Предпорядком $(X, \leq_X)$ называется бинарное отношение $\leq$ на множестве X, такое, что $\forall x, y, z \in X$ имеет место:

– *рефлексивность*: $x \leq x$;

– *транзитивность*: из $x \leq y, y \leq z$ следует $x \leq z$.

Частично упорядоченное множество – предпорядок, в котором выполняется свойство антисимметричности, гласящее, что из $x \leq y, y \leq x$ следует $x = y$.

Предпорядок может быть превращен в частично упорядоченное множество путем уравнивания объектов таких, что $x \leq y, y \leq x$.

Как явно следует из предыдущего раздела, предпорядки могут быть представлены ориентированными графами, в которых узлы описывают элементы предпорядка, а ребра – отношения между элементами. При этом между любыми двумя узлами графа может быть не более одного ребра в одном направлении.

Стоит отметить, что приведенное выше отношение зависимости событий $\rightarrow$ строго аксиоматически предпорядком не является, так как не рефлексивно. Произвольное отношение $\Rightarrow$ напротив задает предпорядок, в чем можно убедиться, найдя соответствия между группами аксиом. Связь между предпорядками и событийными графами лежит на поверхности. Вторые могут быть получены из первых путем удаления отношений рефлексивности $p \leq p$ и циклов.

Монотонное отображение между предпорядками $(A, \leq_A)$ и $(B, \leq_B)$ – функция $f : A \rightarrow B$ такая, что для $\forall x, y \in X$ если $x \leq_A y$ то $f(x) \leq_A f(y)$.

Верхняя грань элемента предпорядка $(P, \leq_P)$:

$$\uparrow p = \{p' \in P \mid p \leq p'\}, p, p' \in P.$$

Верхняя грань предпорядка $(P, \leq_P)$ – подмножество $U \subseteq P$ такое, что если

$$p \in U, p \leq_P q \Rightarrow q \in U.$$

Обозначим $U(P)$ множество верхних граней предпорядка P. Верхние грани предпорядка могут быть в свою очередь упорядочены. Обозначим $U \leq_A V$, если $U \subseteq V$.

В системах управления верхней гранью события является фактически рассмотренное в разделе множество *Eff*(*e*) (эффект события) – множество событий, которое может быть вызвано событием *e*.

Аналогично верхней грани элемента может быть рассмотрена нижняя грань элемента.

Следует отметить, что вследствие рефлексивности отношения предпорядка как верхняя, так и нижняя грань элемента предпорядка (т. е. контекст) однозначно характеризуют элемент. Этот результат является частным случай леммы Йонеды известной в теории категорий при формулировании его для предпорядков [8]. Указанная лемма гласит, что следующие утверждения равносильны:

1) Множество $\uparrow p = \{p' \in P \mid p \leq p'\}$ – верхняя грань порядка P для любого $p \in P$.

2) Отображение $\uparrow: P^{op} \to U(P)$ – монотонно.

3) $p \leq p'$ в P тогда и только тогда, когда $\uparrow(p') \subseteq \uparrow(p)$.

Это утверждение позволяет предположить возможность восстановления структуры графа зависимостей событий при анализе журналов. Наблюдаемая вскоре после возникновения события последовательность других событий однозначно характеризует его. Отношение включения между верхними гранями, по сути, и является тем шаблоном (паттерном), который позволяет установить взаимосвязь между событиями.

Однако, теоретически обоснованные положения не всегда легко реализуются на практике. События в телекоммуникационной сети фиксируются в виде последовательной записи в журнале мониторинга. В процессе регистрации могут возникать ситуации «наложения» граней, когда несколько отказов происходят одновременно, а также появляться случайные «шумы» в виде дублирующихся или отсутствующих событий. Иными словами, возможно взаимное перекрытие независимых событий и вариации во времени наступления их последствий.

Другой проблемой является ограничение по времени наблюдения паттерна события. Согласно лемме Йонеды, паттерны событий для предпорядка представляют собой систему вложенных конечных множеств различной мощности. Наиболее очевидным подходом является разбиение журнала на интервалы наблюдения для каждого события. Важно, чтобы «паттерн» события полностью попал в свой интервал наблюдения.

Анализ событий, основанный на рассмотренном частном случае леммы Йонеды, позволяет, по сути, исключить временной фактор наступления события, сосредоточившись исключительно на их паттернах. Это частично упрощает (абстрагирует) задачу анализа, позволяя оперировать с ограниченным, и, главное, неупорядоченным контекстом. Такой подход также позволяет заменить концепцию отождествления параллелизма с чередованием событий в журналах, как это делается, например, в алгоритмах извлечения процессов (process

mining), на концепцию взаимной вложенности (то есть эквивалентности) граней.

Вопрос о том, какой «хвост» последующих событий необходимо включать в грань, решается с помощью внешних соглашений – например, путем рассмотрения (мульти)множеств последующих событий до окончания кейса. В случае моделирования отказов телекоммуникационных систем можно использовать ранее сформулированное наблюдение о том, что события отказов сетевых элементов в последовательном журнале сопровождаются симметричными им событиями восстановления. При анализе журнала можно рассматривать подпоследовательности событий от отказа до восстановления. Используя верхние грани элементов, можно рассматривать неупорядоченные множества, при этом отношение следования событий определяется включением граней друг в друга. Однако даже в этом случае остаются проблемы наложения граней и шумов.

Некоторые из идей предложенного в работе алгоритма были заимствованы из алгоритма построения ассоциативных правил ECLAT [4].

Транзакциями будем называть все последовательные наборы событий от событий отказа до событий восстановления сетевого элемента. В каждом наборе первое событие является выделенным (т. е. это то событие, ограниченной верхней гранью которого является рассматриваемая транзакция). Важно отметить, что транзакция не является синонимом грани. Последняя – это понятие, относящееся к модели отказов, в то время как транзакция – конкретное наблюдение в журнале событий, которое может быть «зашумлено» присутствием других событий.

Общая схема предложенного алгоритма состоит из следующих этапов:

1) Построение на основе транзакций граней событий;

2) Вычисление матрицы достоверности включения граней;

3) Построение и фильтрация графа зависимостей событий.

Рассмотрим приведенные этапы более подробно.

1) *Построение граней событий*.

Обозначим: E – множество событий отказов, L – входной журнал.

Приведем псевдокод этапа построения граней:

```
EDGE-EXTRACT(E, L, Supp):
    FOR e IN E:
        {Trans} = TRANS_EXTRACT(e, L)
        {F_e} = CALC_FREQS({Trans})
        Tree = BUILD_SUFFIX_TREE({Trans}, {F_e})
        {Edges}=WALK_EDGES(Tree)
        {E,{Edges}}={E,{Edges}} + {Edges}
```

В псевдокоде для экономии места опущены элементарные функции:

– TRANS_EXTRACT(e, L) – формирует множество списков действий (транзакций) журнала L от каждого события отказа e до ближайшего соответствующего ему события восстановления (также типа e);

– CALC_FREQS ({*Trans*}) – вычисляет частоты события на множестве граней {*Trans*};
– WALK_EDGES (*Tree*) – функция, формирующая путем обхода от корня дерева до каждого листа множество путей, соединяющих корень и листья. Каждый путь в таком дереве соответствует грани.

Входным параметром алгоритма является допустимая поддержка (*Supp*) событий в транзакциях, что позволяет регулировать степень фильтрации нечастых событий, вводя в алгоритм механизм абстрагирования.

На первом этапе входной журнал отказов L разбивается на множество транзакций: от события отказа до ближайшего следующего за ним события восстановления. Затем все полученные транзакции упорядочиваются по событиям. Для каждого события в транзакции вычисляется частота его появления. После этого события упорядочиваются по вычисленной поддержке (support), которая определяется как отношение частоты события к общему числу транзакций.

В общем случае поддержка определяется как частота случаев, при которых определенное подмножество событий встречается в наблюдаемых транзакциях:

$$Supp(\uparrow a) = \frac{\#(\uparrow a)}{\sum_{e} \#(\uparrow e)},$$

где $\#(\uparrow e)$ – число транзакций грани события e.

События с недостаточной поддержкой отбрасываются, и формируется список частых событий (F_e). Исходя из принципа монотонности, наиболее частые события встречаются не реже, чем события их вызвавшие, поэтому следует ожидать «связности» полученных множеств событий. Затем производится построение дерева событий для каждой грани. В каждой транзакции осуществляется поиск событий из списка F_e, начиная с самого частого. Если следующее по частоте событие из списка присутствует в транзакции, но отсутствует в текущем дереве, создается новый потомок текущей вершины с с установкой счетчика частоты в 1. В противном случае производится инкремент счетчика уже добавленной вершины. После анализа всех транзакций для каждого дерева формируется множество граней, представляющее собой всевозможные пути в дереве от корня до листьев. Для каждой грани определяется поддержка, равная минимальной поддержке события, входящего в эту грань.

2) *Вычисление матрицы достоверности* включения граней. Для вычисления будем использовать алгоритм, который можно описать следующим псевдокодом:

```
REL-MATRIX(E,{Edges},C):
FOR (e1,{Edges}) IN {E,{Edges}}
        {Edges1} ={Edges1} + { e + {Edge}}
SORT ({Edges1})
FOR (e1,{Edges}) IN {E,{Edges}}
        FOR (e2,{Edges}) IN {E,{Edges}}
                IF Edge_e1 IN Edge_e2:
```

$$Conf(e_1, e_2) = \max\{ Supp(Edge_{e2}) / Supp(Edge_{e1})\}$$
IF $Conf(e_1, e_2) > C$
$$\text{CMATR}[e_1][e_2] = Conf(e_1, e_2)$$

На первом этапе инициирующие события включаются в грани, которые затем упорядочиваются по длине. Начиная с самых коротких граней производится поиск включения этой грани (включая событие-инициатор) в большую грань. При наличии отношения включения производится расчет отношения *достоверности* включения граней друг в друга, которое определяется как условная вероятность обнаружения *a*-грани в *b*-грани как отношение частоты родительской грани к частоте подграни:

$$Conf(\uparrow a \subset \uparrow b) = \frac{\#(\uparrow a \subset \uparrow b)}{\#(\uparrow a)}.$$

Более частая по принципу монотонности подгрань в данном случае присутствует в знаменателе. Иными словами производится вычисление условной вероятности наблюдения родительской грани при наличии в транзакциях журнала подграни.

Вычисленные значения достоверности сводятся в матрицу *Conf* достоверности включения граней. Значения с недостаточным уровнем достоверности отбрасываются.

Дополнительные факты о возможном включении граней (и зависимости событий) можно получить из метрик зависимости и убедительности [10] при некотором увеличении числа параметров алгоритма.

Характеристика зависимости представляет собой отношение частоты совместного наблюдения в журнале граней *a* и *b* к произведению их частот (независимое поведение):

$$Lift(\uparrow a \subset \uparrow b) = \frac{Supp(\uparrow a \subset \uparrow b)}{Supp(\uparrow b) \cdot Supp(\uparrow a)}.$$

В случае, если ее значение больше 1 можно говорить о сильной связи между гранями, если она равна 1, то грани независимы.

Характеристика убедительности оказывает ошибочность включения одной грани в другую. Ошибочность выражается соотношением:

$$Conv(\uparrow a \subset \uparrow b) = \frac{1 - Supp(\uparrow b)}{1 - Conf(\uparrow a \subset \uparrow b)}.$$

Обоснованность включения тем выше, чем больше указанное соотношение (больше 1).

3) *Построение и фильтрация графа* зависимостей событий. Будем считать, что событие *b* связано с событием *a*, если какая-либо грань события *a* вместе с этим событием *a* входит в какую-либо грань события *b* при заданном уровне достоверности. Обозначим этот факт как $a \rightarrow b$.

Два события *a* и *b* будем считать параллельными и обозначать как $a \parallel b$, если $a \rightarrow b$ и $b \rightarrow a$ и $|Conf(a \rightarrow b) - Conf(b \rightarrow a)| < delcon$, где *delcon* – заданный порог параллельности. Действительно, если два события *a* и *b* параллельны, то в журнале с равной частотой возможно наблюдение как транзакций,

в которых перед событием a следует b, так и транзакций, в которых перед b следует a. На этом основана текущая концепция определения параллелизма в алгоритмах извлечения процессов. Однако, в синхронных технических системах данная парадигма не всегда верна. Если два параллельных независимых события имеют, например, нормальное распределение вероятностей времени возникновения от некоторого общего события с достаточно отстоящими друг от друга средними значениями, то в журнале со значительно большей вероятностью будет наблюдаться событие с меньшим средним временем возникновения.

Указанная проблема, в общем случае неразрешимая в физических системах, нивелируется возможностью равновероятного выхода из строя устройств и наблюдения подграней без родительских граней с примерно равными частотами.

Так как формализм графов зависимостей событий предполагает скрытый параллелизм событий при «развилках», важно, чтобы все отношения $a \parallel b$ были удалены.

Экспериментальные исследования

Предложенный алгоритм был реализован и апробирован в следующем серии следующих вычислительных экспериментов. На основе предварительно заданной модели зависимости событий с определенными по условиям эксперимента вероятностями возникновения и распределениями вероятностей длительности между связанными событиями в режиме имитации был сгенерирован журнал отказов телекоммуникационной инфраструктуры. Согласно описанной выше модели поведения, всякий отказ, возникнув, вызывает каскадный отказ связанных с ним устройств (согласно исходному графу зависимостей).

Сгенерированный журнал подавался на вход предложенного алгоритма. Реконструированная модель отказов сравнивалась с изначальной согласно расстоянию редактирования графов [11]. Эксперимент проводился для различных распределений вероятностей длительности между событиями и различных вероятностей отказов сетевых элементов.

В качестве входных параметров модели, помимо графа зависимости событий, состоящего в среднем из 30 узлов, задавался уровень поддержки (support) от 0.05 до 0.9, обеспечивая тем самым различные уровни «агрессивности» алгоритма при реконструкции графа. Для каждого набора параметров проводилась серия из 30 экспериментов. Результаты усреднялись. Было проведено 5 серий экспериментов для различных уровней вероятностей наложения: 0.05; 0.2; 0.4; 0.6; 0.8. В результате оценивались 2 параметра: *точность* реконструкции всех ребер, представленных в оригинальной модели:

$$\alpha_1 = \frac{|E_{rec}|}{|E_{orig}|},$$

где $|E_{orig}|$ – общее число ребер исходной модели, $|E_{rec}|$ – число ребер исходной модели, попавших в реконструированную модель.

Во-вторых, оценивалась *избыточность* – степень наличия в реконструированной модели ребер, не представленных в оригинальной модели:

$$\alpha_2 = 1 - \frac{|E_{new}|}{|E_{orig} \cup E_{new}|},$$

где $|E_{orig} \cup E_{new}|$ – число элементов объединенного множества ребер исходной и реконструированной модели, а $|E_{new}|$ – число ребер реконструированной модели, не присутствовавших в оригинальной модели. Результаты экспериментов представлены на рис. 2–6.

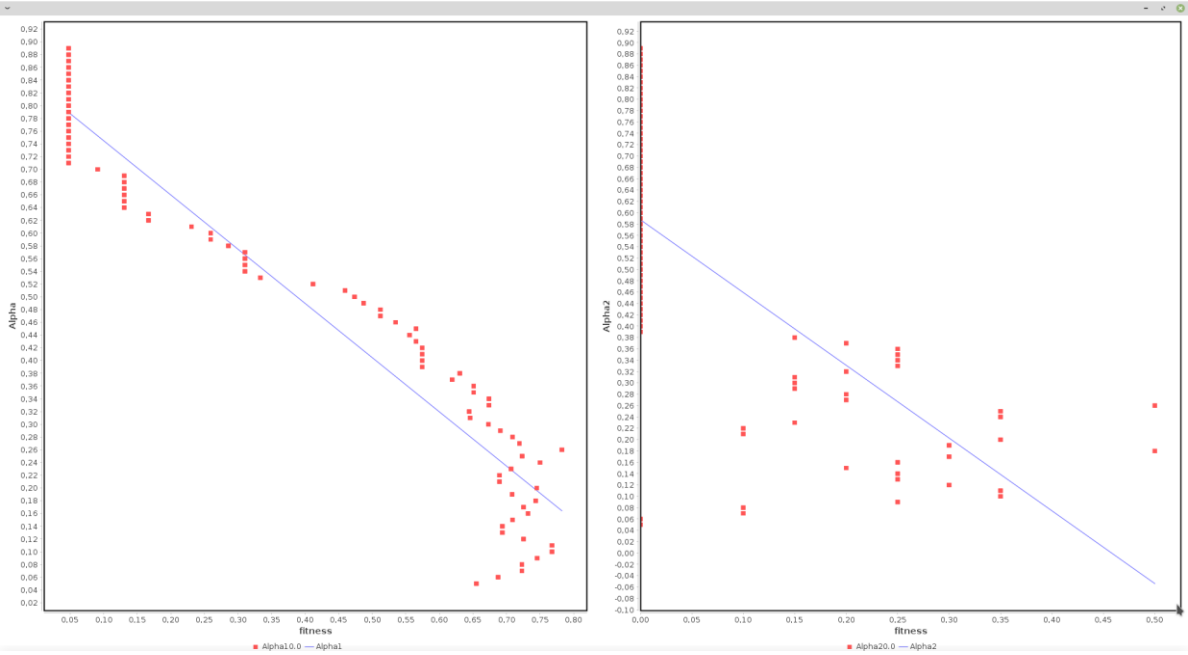

Рис. 2. Зависимость α_1 и α_2 от параметра поддержки для p = 0,05

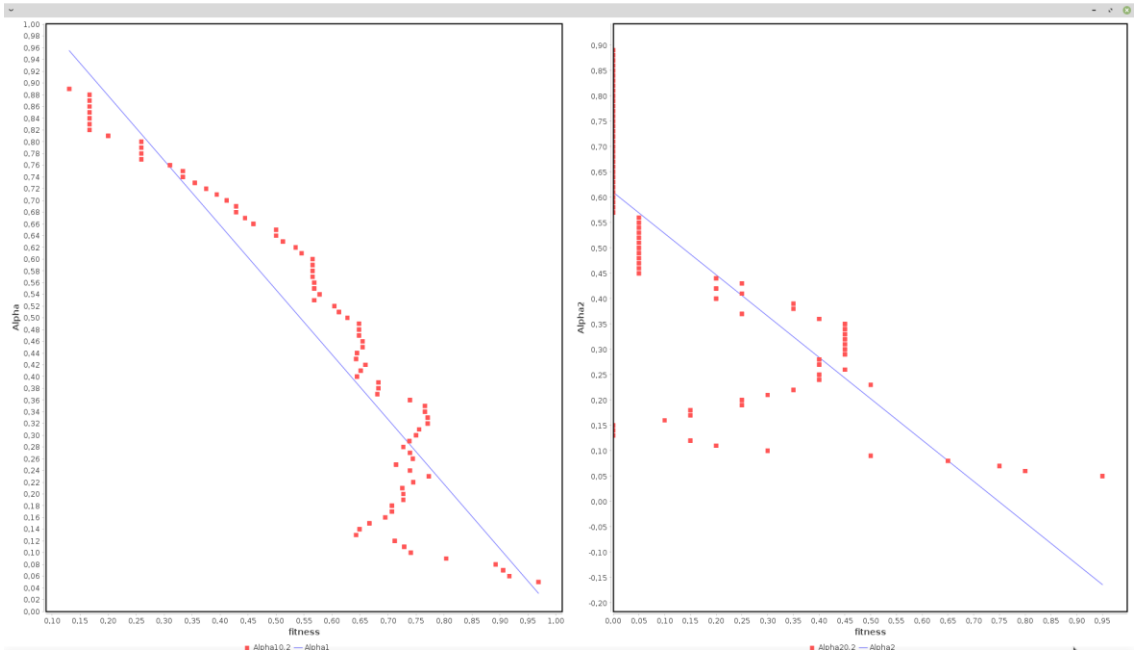

Рис. 3. Зависимость α_1 и α_2 от параметра поддержки для p = 0,2

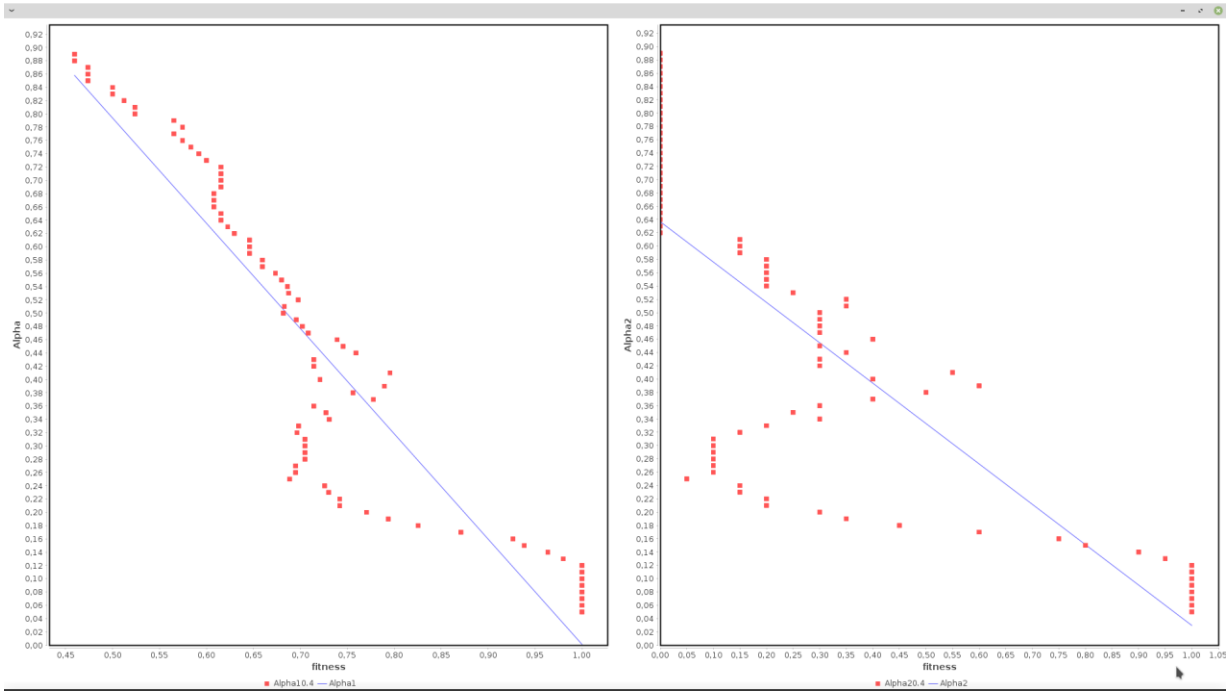

Рис. 4. Зависимость α_1 и α_2 от параметра поддержки для p = 0,4

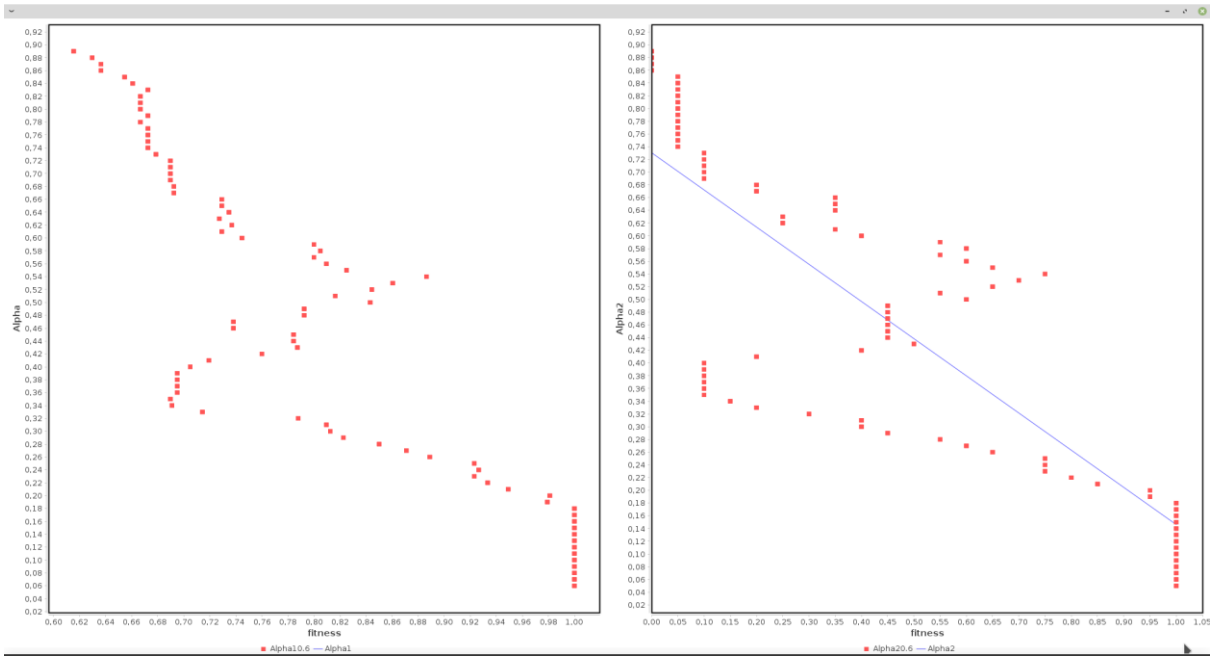

Рис. 5. Зависимость α_1 и α_2 от параметра поддержки для p = 0,6

На основе анализа представленных графиков, отображающих зависимость метрик α_1 (точность реконструкции) и alpha2 (отсутствие лишних ребер) от параметра поддержки при различных уровнях вероятности наложения событий p, можно сделать следующие совокупные выводы:

1) Влияние вероятности наложения p на работу алгоритма:

– низкий уровень наложения (p=0,05): Алгоритм демонстрирует наилучшие результаты. Увеличение поддержки приводит к ухудшению точности (снижение α_1) и улучшению степени отсутствия лишних ребер (некоторый рост α_2), что позволяет найти оптимальный баланс;

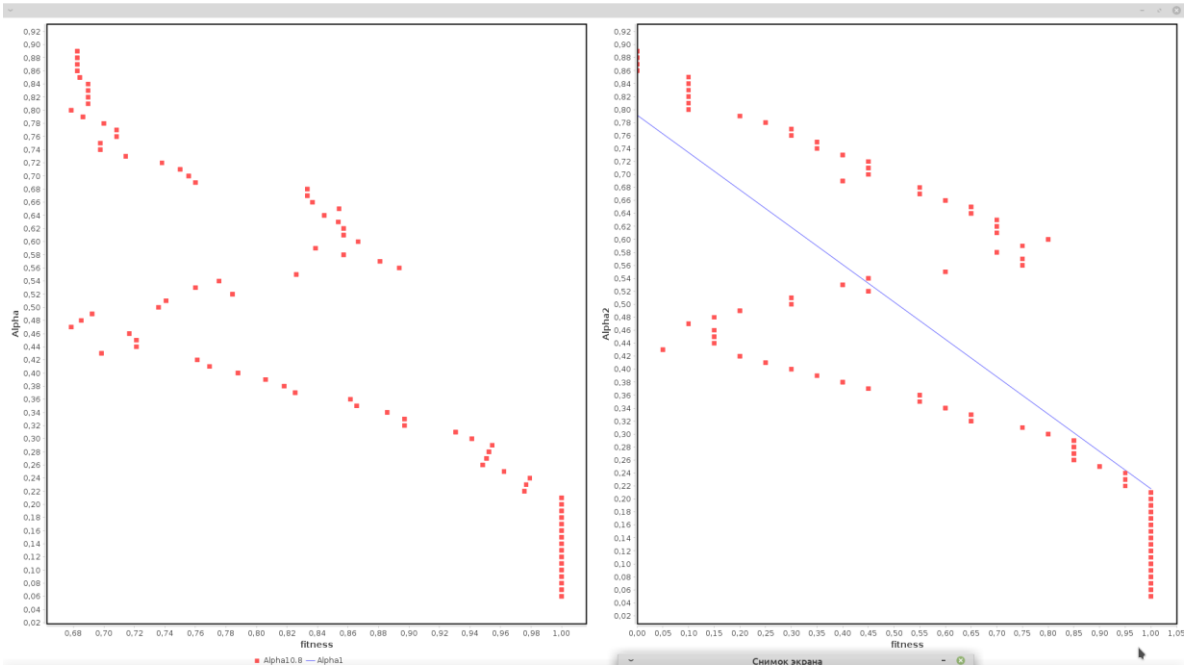

Рис. 6. Зависимость α_1 и α_2 от параметра поддержки для p = 0,8

– умеренный уровень наложения (p = 0,2 и p = 0,4): общая тенденция сохраняется, однако значения α_1 и а α_2 становятся хуже по сравнению с p = 0,05. Найти оптимальный баланс между точностью и отсутствием лишних ребер становится сложнее;
– высокий и очень высокий уровни наложения (p = 0,6 и p = 0,8): алгоритм практически теряет способность к адекватной реконструкции графа зависимостей событий. Зависимости α_1 и α_2 от поддержки становятся слабо выраженными и хаотичными.

2) Влияние параметра поддержки:
– в целом, увеличение поддержки имеет тенденцию к снижению точности реконструкции (α_1), особенно при низких значениях p. При высоких значениях p эта зависимость нивелируется;
– влияние поддержки на степень отсутствия лишних ребер (α_2) менее выражено и более нестабильно. При высоком уровне наложения рост поддержки может приводить к снижению степени отсутствия лишних ребер (появлению ложных связей).

3) Эффективность алгоритма в зависимости от сложности данных:
– алгоритм хорошо работает с данными, имеющими низкий уровень наложения событий. В этих условиях можно найти оптимальный порог поддержки, обеспечивающий компромисс между точностью и степенью исключения «шума»;
– при увеличении вероятности наложения работа алгоритма ухудшается. При p > 0,5 он становится неприменимым для анализа;
– для журналов с высоким уровнем наложения событий необходимо применять методы предварительной обработки данных, направленные на снижение этого наложения. Это может включать в себя фильтрацию дублирующихся событий, объединение близких по времени событий в

кластеры или использование других методов, учитывающих временные зависимости.

Заключение

В работе предложен алгоритм автоматического построения графов зависимостей событий отказов телекоммуникационной сети на основе методов анализа ассоциативных правил. Научная новизна подхода заключается в использовании концепции верхних граней предпорядков для выявления скрытых причинно-следственных связей между событиями, что позволяет в условиях шума и наложения трасс независимых событий повысить точность формирования моделей по сравнению с классическими методами на основе частот отношений непосредственного следования между соседними событиями.

Использование леммы Йонеды позволяет перейти от сложного анализа временных зависимостей к более простому анализу множеств последующих событий. Однако практическая реализация требует аккуратного решения проблем, связанных с наложением граней, шумом и ограничением по времени наблюдения. Дальнейшие исследования должны быть направлены на разработку алгоритмов, которые устойчивы к этим проблемам и эффективно выявляют паттерны событий в реальных телекоммуникационных сетях. Во-вторых, очевидно дублирование в вычислениях суффиксных деревьев при вычислении вложенных граней.

Экспериментальные исследования показали, что разработанный алгоритм демонстрирует высокую эффективность при низком и умеренном уровне наложения событий ($p \leq 0.4$), обеспечивая приемлемый баланс между точностью реконструкции графа (α_1) и степенью отсутствия ложных связей (α_2). Однако при высоком уровне наложения ($p \geq 0.6$) алгоритм теряет устойчивость и требует применения дополнительных методов предобработки данных.

Перспективным направлением дальнейших исследований является интеграция предложенного алгоритма с методами временного анализа событий, кластеризацией последовательностей и адаптивной фильтрацией шумов, что позволит повысить его применимость в условиях сложных и динамичных телекоммуникационных систем.

Литература

1. Nygate Y. A. Event correlation using rule and object based techniques // Integrated Network Management. 1995. P. 278–289.

2. Jakobson G., Weissman M. D. Alarm correlation // IEEE Network. 1993. vol. 7 (6). P. 52–59.

3. Yemini S. A., Kliger S., Mozes E., Yemini Y., Ohsie D. High speed and robust event correlation // IEEE Communications Magazine. 1996. Vol. 34. No. 5. P. 82–90. DOI: 10.1109/35.492975

4. Steinder M., Sethi A. S. Multi-domain diagnosis of end-to-end service failures // Hierarchically Routed Networks (Lecture Notes in Computer Science). 2004. Vol. 3042. P. 1036–1046. DOI: 10.1007/978-3-540-24693-0_85

5. Hasan M., Sugla B., Viswanathan R. A conceptual framework for network management event correlation and filtering systems // Proc. 6th IFIP/IEEE Int. Symp. on Integrated Network Management (IM'99). Boston, MA, USA, May 1999. P. 233–246.

6. Vilalta R., Ma S. Predicting rare events in temporal domain // Proc. IEEE Int. Conf. on Data Mining (ICDM). 2002. P. 474–481. DOI: 10.1109/ICDM.2002.1183991.

7. Steinder M., Sethi A. S. Increasing robustness of fault localization through analysis of lost, spurious, and positive symptoms // Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM). New York, NY, USA, 2002. Vol. 1. P. 322–331. DOI: 10.1109/INFCOM.2002.1019274

8. Fong B., Spivak D. I. An Invitation to Applied Category Theory. Seven Sketches in Compositionality. Cambridge: Cambridge University Press, 2019. 348 p.

9. Карпов Ю. Г. Model Checking. Верификация параллельных и распределённых программных систем (+CD-ROM). – СПб.: БХВ-Петербург, 2010. – 552 с.

10. Zaki M. J. Scalable algorithms for association mining // IEEE Transactions on Knowledge and Data Engineering. 2000. Vol. 12. No. 3. P. 372–390.

11. Будко Н. П., Васильев Н. В. Обзор графо-аналитических подходов к мониторингу информационно-телекоммуникационных сетей и их применение для выявления аномальных состояний // Системы управления, связи и безопасности. 2021. № 6. С. 53–75. DOI: 10.24412/2410-9916-2021-6-53-75

References

1. Nygate Y. A. Event correlation using rule and object based techniques. *Integrated Network Management*, 1995, pp. 278–289.

2. Jakobson G., Weissman M. D. Alarm correlation. *IEEE Network*, 1993, vol. 7 (6), pp. 52–59.

3. Yemini S. A., Kliger S., Mozes E., Yemini Y., Ohsie D. High speed and robust event correlation. *IEEE Communications Magazine*, 1996, vol. 34, no. 5, pp. 82–90. DOI: 10.1109/35.492975.

4. Steinder M., Sethi A. S. Multi-domain diagnosis of end-to-end service failures. *Hierarchically Routed Networks (Lecture Notes in Computer Science)*, 2004, vol. 3042, pp. 1036–1046. DOI: 10.1007/978-3-540-24693-0_85.

5. Hasan M., Sugla B., Viswanathan R. A conceptual framework for network management event correlation and filtering systems. *Proceedings of the 6th IFIP/IEEE International Symposium on Integrated Network Management (IM'99)*, Boston, MA, USA, May 1999, pp. 233–246.

6. Vilalta R., Ma S. Predicting rare events in temporal domain. *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2002, pp. 474–481. DOI: 10.1109/ICDM.2002.1183991.

7. Steinder M., Sethi A. S. Increasing robustness of fault localization through analysis of lost, spurious, and positive symptoms. *Proceedings of the Twenty-First*

Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), New York, NY, USA, 2002, vol. 1, pp. 322–331. DOI: 10.1109/INFCOM.2002.1019274.

8. Fong B., Spivak D. I. *An Invitation to Applied Category Theory. Seven Sketches in Compositionality.* Cambridge, Cambridge University Press, 2019. 348 p.

9. Karpov Yu. G. *Model Checking. Verifikatsiia parallel'nykh i raspredelennykh programmnykh sistem* (+ *CD-ROM*) [Model Checking. Verification of Parallel and Distributed Software Systems (+ CD-ROM)]. Saint Petersburg, BKhV-Peterburg Publ., 2010. 552 p. (in Russian).

10. Zaki M. J. Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, 2000, vol. 12, no. 3, pp. 372–390.

11. Budko N. P., Vasiliev N. V. Obzor grafo-analiticheskikh podkhodov k monitoringu informatsionno-telekommunikatsionnykh setei i ikh primenenie dlia vyiavleniia anomal'nykh sostoianii [Review of Graph-Analytical Approaches to the Monitoring of Information and Telecommunication Networks and Their Application for the Detection of Abnormal States]. *Systems of Control, Communication and Security*, 2021, no. 6, pp. 53–75. DOI: 10.24412/2410-9916-2021-6-53-75 (in Russian).

Статья поступила 19 октября 2025 г.

Сведения об авторах

Васильев Николай Владимирович – кандидат технических наук, доцент. Доцент кафедры информационных систем. Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина). Область научных интересов: управление сетями связи, анализ бизнес-процессов. E-mail: gandvik1984@gmail.com

Адрес: 197022, Россия, г. Санкт-Петербург, ул. Проф. Попова, д. 5, лит. Ф.

Буравлев Андрей Сергеевич – аспирант. Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина). Область научных интересов: методы обработки естественного языка. E-mail: rublick321@mail.ru

Адрес: 197022, Россия, г. Санкт-Петербург, ул. Проф. Попова, д. 5, лит. Ф.

Будко Никита Павлович – кандидат технических наук. Независимый специалист. Область научных интересов: мониторинг информационных ресурсов; сбор и обработка информации. E–mail: budko62@mail.ru

Адрес: 194064, г. Санкт-Петербург, ул. Бутлерова, 9, корп. 3, кв. 252.

Algorithm for building a correlation model of telecommunication network faults based on Yoneda's lemma and the ECLAT algorithm

N. V. Vasiliev, A. S. Buravlev, N. P. Budko

Problem Statement. *Modern telecommunication networks generate massive volumes of event data, within which failures and warnings constitute critical categories. Effective correlation of such events is a key prerequisite for timely diagnosis and elimination of malfunctions. The relevance of this research stems from the need to develop methods capable of automatically detecting causal relationships among events in environments characterized by high complexity, noisy data overlays, and concurrent processes within distributed infrastructures. The* ***objective*** *of this study is to construct a formal model and algorithm for the automatic reconstruction of dependency graphs of network element failure events based on the analysis of monitoring logs. The work presents a description of the formalism of event dependency graphs, introduces a proposed algorithm for graph construction using suffix trees, and provides a brief overview along with computational experiments assessing the impact of noise on the accuracy of model reconstruction by the implemented prototype. The proposed algorithm comprises the following stages: (i) construction of event facets, (ii) formation of a relation matrix between these facets, and (iii) filtering and assembly of the final model. Event facets are derived using* ***ECLAT suffix trees*** *applied to event log transactions in the monitoring system. Transactions are identified through a heuristic based on the symmetry between failure events and corresponding recovery events. At the next stage, conditional probabilities of facet inclusions are analyzed, taking into account possible concurrency of events, with low-probability relations being filtered out. Finally, the resulting relation model between facets is transformed into an event dependency graph. The* ***methodological*** *foundation of this research includes: computational graph theory; data analysis methods; probability theory and mathematical statistics; systems analysis; systems modeling theory; set theory; and category theory. The* ***scientific novelty*** *of the proposed approach lies in the developed concept of event dependency based on the inclusion of least upper facets of preorder elements. This enables the construction of event dependency graphs under conditions of noise and overlapping event chains. The* ***practical outcome*** *of the study is the ability to automate the analysis of event logs in telecommunication networks, thereby reducing the workload of support personnel and accelerating the identification of root causes of failures. Experimental results demonstrate that the proposed method is effective under low and moderate levels of event overlap ensuring high accuracy of dependency reconstruction. Thus, this research lays the foundation for the development of intelligent monitoring systems capable of uncovering hidden patterns and supporting the reliability of distributed information and telecommunication infrastructures.*

Key words: *associative rules, event dependence graphs, correlation and filtering of events, Ioneda's lemm, partially ordered sets, process extraction.*

About the Authors

Nikolai Vladimirovich Vasiliev – PhD, Associate Professor, Department of Information Systems. Saint Petersburg Electrotechnical University "LETI" named after V. I. Ulyanov (Lenin). Field of research: communication network management, business process analysis. E-mail: gandvik1984@gmail.com

Address: 197022, Russia, St. Petersburg, Professor Popov str., build. 5, lit. F.

Andrey Sergeevich Buravlev – Postgraduate Student. Saint Petersburg Electrotechnical University "LETI" named after V. I. Ulyanov (Lenin). Field of research: natural language processing methods. E-mail: asburavlev@stud.etu.ru

Address: 197022, Russia, St. Petersburg, Professor Popov str., build. 5, lit. F.

Nikita Pavlovich Budko – PhD. Of Engineering Sciences. An independent expert. Field of research: information resource monitoring; data collection and processing. Email: budko62@mail.ru

Address: 194064, Saint Petersburg, Butlerova str., 9/3.