

УДК 004.415.2:004.932.2
DOI 10.17513/snt.40895



CC BY 4.0

ПРОГРАММНАЯ МОДЕЛЬ ТРАССИРУЕМОГО КОНТУРА ВИДЕОАНАЛИТИКИ С КОНТРОЛЕМ ПРОИСХОЖДЕНИЯ РЕЗУЛЬТАТОВ

^{1,2}Белкин Е. Н. ORCID ID 0009-0007-5848-5433

¹Московский государственный технический университет имени Н. Э. Баумана
(национальный исследовательский университет), Калуга, Российская Федерация;

²Акционерное общество «АЛГОНТ», Калуга, Российская Федерация,
e-mail: belkinen@algont.ru

Интеллектуальные системы видеонаблюдения формируют метаданные, используемые при операторских и автоматизированных решениях. Точность модели не показывает, какой версией компонента получен результат, при каких настройках и состоянии среды он сформирован и с каким архивным фрагментом связан. Цель исследования – разработать и экспериментально проверить программную модель трассируемого контура видеоаналитики, контролирующую происхождение, конфигурацию и эксплуатационную пригодность каждого результата. Результат представлен как версионизируемый объект с идентификаторами источника и детектора, версией и контрольной суммой модели, контрольной суммой конфигурации, показателями качества кадра, профилем среды, ссылками на архив и аудит и состоянием операторского подтверждения. Сформулированы контракт «проверка предусловий – вычисление – связывание – аудит – реакция» и двухуровневый критерий доверия, разделяющий техническую целостность и эксплуатационную допустимость. Проверка выполнена на 360 видеосценариях семи типов; профили сравнивались на одинаковых потоках при фиксированном аналитическом стеке. Качество отдельных моделей не являлось переменной эксперимента. Предлагаемый контур повысил полноту привязки событий к версии модели, конфигурации, качеству входа и архиву, сократил время операторского подтверждения и снизил долю неverified реакций. Полученные результаты демонстрируют реализуемость модели в пределах испытанной конфигурации и отделяют программный жизненный цикл события от характеристик движка распознавания.

Ключевые слова: видеоаналитика, программная архитектура, трассируемость результата, управление конфигурацией, качество изображения, машинное обучение

SOFTWARE MODEL OF A TRACEABLE VIDEO ANALYTICS PIPELINE WITH RESULT PROVENANCE CONTROL

^{1,2}Belkin E. N. ORCID ID 0009-0007-5848-5433

¹Bauman Moscow State Technical University (National Research University),
Kaluga, Russian Federation;

²ALGONT Joint Stock Company, Kaluga, Russian Federation,
e-mail: belkinen@algont.ru

Intelligent video surveillance systems generate metadata used in operator and automated decisions. Model accuracy does not show which component version produced a result, under which settings and runtime state it was generated, or how it is linked to an archive fragment. The study aimed to develop and experimentally evaluate a software model of a traceable video analytics pipeline controlling the provenance, configuration, and operational fitness of each result. A result is represented as a versioned object containing source and detector identifiers, model and configuration checksums, frame-quality indicators, the runtime profile, archive and audit references, and operator-confirmation state. A processing contract comprising precondition checking, computation, binding, auditing, and reaction was formulated together with a two-level trust criterion separating technical integrity from operational acceptability. The evaluation used 360 video scenarios of seven event types; the profiles were compared on identical streams with the analytics stack held fixed. Recognition quality of individual models was not an experimental variable. The proposed pipeline increased the completeness of binding events to model version, configuration, input quality, and archive, reduced operator-confirmation time, and decreased unverified reactions. The findings demonstrate feasibility within the tested configuration and separate the event lifecycle from the characteristics of a particular recognition engine.

Keywords: video analytics, software architecture, result traceability, configuration management, image quality, machine learning

Введение

Современный программный комплекс видеоаналитики формирует не только изображение, но и производные объекты: события, координаты, классы, траектории, оценки уверенности и операторские решения. В прикладной системе именно эти

объекты запускают поиск в архиве, уведомление и регламент реагирования. Поэтому доверие к результату нельзя сводить к одной числовой оценке нейросетевой модели: необходимо знать происхождение данных, состояние программных компонентов и возможность последующей проверки.

В российских исследованиях доверенного искусственного интеллекта эта проблема рассматривается как совокупность доверия к данным, модели, процедурам применения и цифровой среде [1; 2].

Практики эксплуатации моделей машинного обучения и программной инженерии требуют версионирования данных и моделей, автоматизированных проверок и наблюдаемости [3-5]. Рубрики готовности, паспорта моделей и механизмы обработки отказов добавляют сведения об области применения, метриках, ограничениях и деградации [6-8]. Однако их основной объект – модель или конвейер, а не отдельное событие, непосредственно передаваемое оператору или исполнительной подсистеме.

Для видеопотока особенно важна зависимость результата от качества входного кадра. Выбор пригодного кадра и формализация показателей качества позволяют сохранять связь результата с исходным изображением [9; 10]. Высокие средние показатели детекторов не устраняют зависимость от разрешения, освещенности, ракурса, перекрытий и порога принятия решения [11; 12].

В задачах обнаружения аномалий эта зависимость усиливается из-за редкости событий и неполноты выборок [13; 14]. Кроме того, численная уверенность нейронной сети может быть плохо откалибрована [15]. Поэтому высокая уверенность не должна компенсировать отсутствие сведений о модели, конфигурации, среде исполнения или архивной ссылке.

Происхождение данных и программных артефактов исследуется на нескольких уровнях. Модель PROV формализует сущности, действия, агентов и отношения по-

рождения, использования и производительности [16]. В российской MLOps-платформе НИУ ВШЭ объектная модель вычислительного эксперимента и язык его спецификации применяются для воспроизводимости и прослеживаемости распределенных вычислений [17]. GeneaLog связывает каждое выходное событие потоковой обработки с исходными данными посредством мелкозернистой provenance на уровне отдельных кортежей [18]. Проверяемые журналы обеспечивают обнаружение постфактум-изменений: в работе [19] для записей формируются доказательства существования и целостности, сохраняемые в блокчейне с ограниченным составом участников. Сопоставление этих подходов с предлагаемой моделью приведено в табл. 1.

Предлагаемая модель не заменяет перечисленные подходы, а задает их доменное применение. Объект r_i может быть представлен в PROV как Entity, сформированная Activity инференса; модель, конфигурация, кадр и архивный фрагмент – как связанные Entities, а детектор и оператор – как Agents [16]. Объектная модель MLOps-эксперимента может фиксировать структуру вычисления, конвейеры, наборы данных и выполненные операции [17]; GeneaLog – связывать отдельный выход потока с породившими его входными элементами [18]; проверяемый журнал – подтверждать отсутствие незаметной модификации аудиторских записей [19]. Новизна состоит в обязательной эксплуатационной семантике: описание происхождения само по себе не разрешает реакцию при $G(r_i)=0$, а отклоненный результат сохраняется для аудита.

Таблица 1

Сопоставление моделей происхождения и предлагаемого контура

Подход	Базовая единица	Используемые механизмы	Недостающие свойства для допуска видеособытия
PROV	Entity – Activity – Agent	Граф происхождения, ответственность и обмен метаданными	Нет обязательных полей качества кадра, архива, подтверждения и блокировки реакции
MLOps-платформа НИУ ВШЭ	Вычислительный эксперимент	Объектная модель, спецификация, интеграция конвейеров и данных	Ориентация на эксперимент; нет контракта для каждого результата инференса
GeneaLog	Потоковый кортеж/событие	Мелкозернистая связь выхода с исходными данными	Нет проверки модели, конфигурации, кадра и политики реакции
Проверяемый журнал	Запись аудита	Доказательства существования и целостности, распределенная проверка	Защищает журнал, но не задает семантику результата и допуска к реакции
Настоящая работа	Отдельное событие r_i	$G(r_i)$, $Q(r_i)$, архив, аудит и состояние подтверждения	Доменный контракт; совместим с PROV, MLOps provenance, event-level lineage и tamper-evident logging

Примечание: составлено автором по материалам [16-19].

Нерешенным остается построение исполняемого программного контракта, в котором результат видеоаналитики проверяется на всем пути от кадра до реакции. Рабочая гипотеза состоит в том, что обязательное включение происхождения, конфигурации, среды исполнения, качества входа, архивной ссылки и аудита повышает трассируемость и эксплуатационную пригодность события при неизменном аналитическом стеке.

Научная новизна заключается в сочетании формального объекта результата на уровне отдельного события, жестких предусловий, не компенсируемых высокой уверенностью, и двухуровневого критерия доверия, разделяющего техническую целостность и эксплуатационную допустимость. В отличие от универсальных моделей lineage, предложенная модель задает обязательные поля, состояние

отказа и условие передачи события в контур реакции.

Цель исследования – разработать программную модель трассируемого контура видеоаналитики и проверить, повышает ли ее применение полноту происхождения результатов, управляемость конфигурации и операционную эффективность операторского подтверждения при неизменном составе и настройках аналитического стека.

Материал и методы исследования

Объектом исследования являлся программный конвейер обработки видеособытия, включающий источник, детектор, реестр моделей и конфигураций, среду исполнения, архив, журнал и рабочее место оператора. Единицей анализа выбран отдельный результат r_i , а не агрегированная запись о состоянии модели. Результат задается кортежем:

$$r_i = \langle id_i, t_i, s_i, d_i, v_i, h_i^M, h_i^c, q_i^f, p_i, \rho_i, a_i, j_i, o_i \rangle, \quad (1)$$

где id_i – идентификатор события; t_i – временные метки источника и системы; s_i – источник; d_i – детектор; v_i – версия модели; h_i^M и h_i^c – контрольные суммы модели и конфигурации; q_i^f – вектор показателей качества кадра; p_i – профиль среды исполнения; ρ_i – оценка уверенности; a_i и j_i – ссылки на архив и аудит; o_i – состояние операторского подтверждения.

В модели различаются жесткие и мягкие условия. Жесткий предикат $G(r_i)$ проверяет наличие и согласованность контекста. Индикаторы подтвержденности источника, модели, конфигурации, среды исполнения, времени, архивной ссылки и аудита обозначим через $S_i, M_i, C_i, P_i, T_i, A_i$ и J_i соответственно; тогда:

$$G(r_i) = S_i \wedge M_i \wedge C_i \wedge P_i \wedge T_i \wedge A_i \wedge J_i \quad (2)$$

Если хотя бы одно условие ложно, событие не удаляется, а сохраняется со статусом недоверенного результата и машиночитаемой причиной. Это решение важно для диагностики: отсутствие архивной ссылки или несовпадение контрольной суммы само по себе является наблюдаемым отказом программного контура.

Мягкая оценка $Q(r_i)$ объединяет качество кадра, состояние модели, конфигурации, вычислительной платформы и полноту аудита. Веса задаются эксплуатационной политикой и в сумме равны единице; способ их выбора относится к настройке конкретной системы и не является предметом настоящего исследования:

$$Q(r_i) = \alpha q_i^f + \beta q_i^M + \gamma q_i^c + \delta q_i^p + \varepsilon q_i^A, \quad \alpha + \beta + \gamma + \delta + \varepsilon = 1 \quad (3)$$

Итоговый статус определяется совместно жестким предикатом, порогом q_c и политической подтвержденности H_i :

$$\text{Trusted}(r_i) = G(r_i) \wedge [Q(r_i) \geq q_c] \wedge H_i \quad (4)$$

Значение H_i равно единице автоматически только для событий, для которых это разрешено эксплуатационным регламентом. Для критических событий может требоваться подтверждение оператором либо вторым сенсором. Тем самым технически целый результат остается аналитической подсказкой, если установленная политика требует дополнительной верификации.

Программный контракт обработки реализован как последовательность «проверка предусловий – вычисление – связывание – аудит – реакция». До вычисления проверяются источник, версия модели и профиль среды. После вычисления результат связывается с контрольной суммой конфигурации, качеством кадра и архивом, затем формируется запись аудита с контролем целостности.

Таблица 2

Проверяемые элементы программного объекта результата видеоаналитики

Элемент результата	Проверяемое условие	Поведение при нарушении
Источник и кадр	Идентификатор источника, временная связность, показатели резкости, освещенности и потерь	Сохранение события со статусом <code>untrusted_input</code> ; автоматическая реакция запрещена
Модель	Версия, контрольная сумма, разрешенный паспорт модели и область применения	Блокировка доверенной реакции; регистрация <code>model_mismatch</code>
Конфигурация	Контрольная сумма порогов и зон, полномочия субъекта изменения	Изменение отклоняется до записи; формируется <code>audit_fail</code>
Среда исполнения	Профиль узла, версии библиотек, состояние времени и контроля целостности	Статус <code>untrusted_platform</code> до восстановления профиля
Архив и аудит	Наличие <code>archive_ref</code> и <code>audit_ref</code> , соответствие источнику и временному интервалу	Событие сохраняется, но не используется как доверенное основание
Подтверждение	Оператор или второй сенсор для событий, указанных эксплуатационным регламентом	Результат остается подсказкой без исполнительной команды

Примечание: составлено автором по результатам исследования.

Реакция разрешается только после вычисления итогового статуса. Повтор события с тем же идентификатором и номером последовательности идемпотентен и не создает вторую доверенную запись. Состав проверяемых элементов и реакции на нарушения приведены в табл. 2.

Экспериментальная проверка выполнена на 360 видеосценариях длительностью 655 камера-часов. Испытательный поток включал 1840 положительных событий и 9600 отрицательных временных окон от 27 камер. Рассматривались движение, пересечение линии, саботаж, оставленный предмет, скопление людей, распознавание государственного регистрационного знака и нахождение объекта в запрещенной зоне. Разрешение камер составляло 2 мегапикселя, частота – 25 кадров в секунду; освещенность изменялась от 80 до 500 люкс в помещениях и от 30 до 1000 люкс на открытых участках.

Сценарий представлял ограниченный по времени видеосегмент с идентификатором камеры, ожидаемым типом события и целевой временной отметкой. Положительным считалось событие, совпавшее со сценарным протоколом по типу и времени; отрицательным окном – интервал без целевого события. Ложной тревогой считалось срабатывание в отрицательном окне, а неverified реакцией – его передача в контур реагирования без отметки оператора или второго сенсора. Распределение сценариев/событий: движение 64/368; пересечение линии 58/322; саботаж 42/154; оставленный предмет 44/176; скопление людей 46/201; ГРЗ 55/351; запрещенная зона 51/268. Разметку независи-

мо выполняли два специалиста; 67 (3,6%) разногласий разрешал третий эксперт. Согласие до разрешения разногласий составило 96,4%, коэффициент к Коэна – 0,957 [95% ДИ 0,946; 0,968]. Настроечные фрагменты не входили в оценочные 655 камера-часов. Два программных профиля обрабатывали одинаковые видеопотоки при неизменных составе и конфигурации аналитического стека, версиях модельных артефактов, порогах и среде исполнения. Единицей парного сопоставления являлось одно и то же событие или отрицательное окно, согласованное по идентификатору камеры, исходному времени и типу детектора. Базовый профиль сохранял прикладные журналы и допускал необязательную связь с архивом. Предлагаемый профиль требовал заполнения r_i , выполнения $G(r_i)$, контроля версий и конфигурации, архивной и аудиторской ссылок и фиксации качества входа.

В операторском эксперименте участвовали 8 операторов со стажем 2-9 лет (медиана 4,5 года). Стратифицированно отобраны 560 событий (по 80 каждого типа). В рандомизированном перекрестном дизайне каждое событие оценивалось в обоих профилях разными операторами; каждый обработал по 70 событий на профиль. Получено 560 парных сопоставлений. Время подтверждения определялось от открытия карточки события до фиксации решения; автоматические решения в эту метрику не включались.

Для показателей полноты знаменателем служили 1840 положительных событий, для доли неverified реакций – 9600 отрицательных окон. Дополнительно выполнялись 600 попыток подмены мо-

дельного артефакта или несанкционированного изменения порога и 300 сценариев отката версии или отсутствия архивной ссылки. Первичные признаки сохранялись на уровне события или окна вместе с идентификаторами камеры, сценария, оператора и сессии.

Статистический анализ задавался как парный и иерархический. Для каждого бинарного исхода вычисляли внутриобъектную разность $y_i(\text{предл.}) - y_i(\text{баз.})$ для одного и того же события или отрицательного окна. 95%-ный доверительный интервал средней парной разности оценивали двухступенчатым кластерным bootstrap с 10 000 повторов: камеры ресемплировали с возвращением, затем внутри выбранной камеры с возвращением ресемплировали сценарии при сохранении обеих профильных записей каждой пары. Нулевую гипотезу проверяли точным тестом Макнемара; отдельные доли сопровождали описательными интервалами Уилсона.

В анализе чувствительности бинарные исходы оценивали логистическими смешанными моделями с фиксированным эффектом профиля и случайными интерцептами события, сценария и камеры. Время подтверждения анализировали после логарифмирования в линейной смешанной модели

с фиксированными эффектами профиля, периода и порядка предъявления и случайными интерцептами события, оператора и сессии. Эффект представляли как отношение геометрических средних; разность медиан и ее 95%-ный доверительный интервал дополнительно оценивали перекрестным кластерным bootstrap по событиям и операторам с 10 000 повторов. Все проверки были двусторонними, уровень значимости составлял 0,05. Расчеты выполнены в Python 3.12.7 с пакетами NumPy 2.2.2, SciPy 1.15.1 и statsmodels 0.14.4.

Результаты исследования и их обсуждение

При фиксированном аналитическом стеке анализ был направлен на программный жизненный цикл результата. Показатели наличия версии и конфигурации, архивной ссылки, качества кадра и совместного выполнения $G(r_i)=1$ являются мерами выполнения контракта, поэтому их рост демонстрирует принудительную полноту, но не изменение точности детектора. Операционный эффект оценивали по времени подтверждения и доле отрицательных окон, в которых не верифицированная ложная тревога дошла до реакции. Результаты парного кластерного анализа приведены в табл. 3.

Таблица 3

Влияние обязательной трассировки на эксплуатационные показатели

Показатель	Базовый профиль, n/N; оценка [95% ДИ Уилсона]	Предлагаемый профиль, n/N; оценка [95% ДИ Уилсона]	Парная разность [кластерный 95% ДИ]; p
Версия модели и контрольная сумма конфигурации	810/1840; 0,440 [0,418; 0,463]	1803/1840; 0,980 [0,972; 0,985]	+0,540 [+0,507; +0,571]; p < 0,001
Архивная ссылка	1214/1840; 0,660 [0,638; 0,681]	1748/1840; 0,950 [0,939; 0,959]	+0,290 [+0,253; +0,326]; p < 0,001
Зафиксированное качество кадра	570/1840; 0,310 [0,289; 0,331]	1711/1840; 0,930 [0,917; 0,941]	+0,620 [+0,585; +0,653]; p < 0,001
Совместное выполнение жесткого предиката $G(r_i)=1$	424/1840; 0,230 [0,212; 0,250]	1593/1840; 0,866 [0,849; 0,881]	+0,635 [+0,595; +0,671]; p < 0,001
Время подтверждения оператором, с	n = 560; 21,4; IQR 17,8-26,9	n = 560; 14,2; IQR 11,6-18,5	-7,2 с (-33,6%); [-8,6; -5,9]; p < 0,001
Отрицательные окна с не верифицированной реакцией	1728/9600; 0,180 [0,172; 0,188]	384/9600; 0,040 [0,036; 0,044]	-0,140 [-0,158; -0,121]; p < 0,001

Примечание: IQR – межквартильный размах; 95% ДИ – 95%-ный доверительный интервал. Для бинарных исходов в последнем столбце приведена средняя парная разность с двухступенчатым кластерным bootstrap-интервалом и p-значением точного теста Макнемара; для времени – разность медиан с перекрестным кластерным bootstrap-интервалом.

Оценка производительности

Накладные расходы измеряли без повторного выполнения инференса, поскольку аналитический стек в профилях был одинаковым. На одном узле x86-64 с 8 вычислительными ядрами (AMD Zen4 3,6 ГГц), 32 ГБ оперативной памяти DDR5-5600 в двухканальном режиме и накопителем NVMe PCI-E 4.0 выполнено 30 прогонов по 100 000 событий после 10 000 прогревочных сообщений; контрольные суммы модели и конфигурации вычислялись при загрузке, а записи аудита объединялись в пакеты по 128. При нагрузке 512 событий/с медиана и 95-й процентиль задержки программной обработки составили соответственно 2,18 и 4,76 мс в базовом профиле и 2,55 и 5,34 мс в предлагаемом; абсолютные накладные расходы составили 0,37 и 0,58 мс. Предельная устойчивая пропускная способность снизилась с 3410 до 3110 событий/с (на 8,8%), загрузка процессора при целевой нагрузке выросла с 14,6 до 19,1%, а резидентная память – на 82 МБ. В 3,0 млн обработок потерь событий и расходов между доверенным результатом и записью аудита не зарегистрировано. В испытанной конфигурации сохранялся более чем шестикратный запас по пропускной способности относительно нагрузки 512 событий/с. При переносе журнала в сетевое хранилище и изменении способа пакетирования требуется повторная нагрузочная проверка.

В парном анализе число дискордантных пар «базовый профиль: нет; предлагаемый: да» и «базовый: да; предлагаемый: нет» составило 1005/12 для версии модели и конфигурации, 556/22 для архивной ссылки, 1163/22 для качества кадра и 1183/14 для $G(r_i)=1$. Во всех сравнениях направление эффекта сохранялось после учета камеры и сценария ($p < 0,001$).

Наибольший прирост получен для фиксации качества входа и совместного выполнения жестких условий. Это ожидаемое следствие обязательной схемы: результат не мог получить доверенный статус без соответствующих полей. После учета события, оператора, сессии, периода и порядка предъявления отношение геометрических средних времени подтверждения составило 0,67 [95% ДИ 0,63; 0,72]; эффект порядка был незначим ($p = 0,41$), как и взаимодействие профиля с периодом ($p = 0,58$). Карточка события сразу содержала исходный фрагмент, версию стека, конфигурацию и причину понижения доверия, что уменьшало число переходов между интерфейсами. Поэтому сокращение времени следует

интерпретировать как эффект интегрированного профиля, включавшего контракт трассируемости и измененную подачу контекста, а не как изолированный эффект одного механизма происхождения.

Отрицательные тесты подтвердили работу жестких предусловий. Подмена файла модели и изменение порога без разрешенной роли были заблокированы в 600 из 600 попыток; нижняя граница 95%-ного интервала Уилсона составила 99,4%. Откат модели и отсутствие архивной ссылки были корректно обнаружены в 300 из 300 сценариев; нижняя граница интервала составила 98,7%. При отсутствии архивной ссылки событие сохранялось со статусом *untrusted archive*, что позволяло расследовать отказ без ложного признания результата доверенным. Полученные 100% относятся только к заданному набору интерфейсных и интеграционных тестов и не означают абсолютной защищенности программного комплекса.

Полученные результаты согласуются с практиками MLOps и паспортами моделей [3; 6; 7], но переносят контроль на уровень отдельного события. PROV предоставляет основу для обмена графом происхождения [16], объектные модели вычислительных экспериментов – для воспроизводимости и прослеживаемости MLOps-конвейеров [17], а GeneaLog показывает реализуемость мелкозернистой provenance на уровне выходных событий потоковой обработки [18]. Проверяемое журналирование усиливает обнаружение изменений аудиторских записей после их формирования [19]. Контракт концептуально не зависит от архитектуры распознавания, однако величина эффекта при смене стека требует отдельной проверки.

При межсистемной интеграции отображение выполняется без потери обязательной семантики. Идентификаторы id_i , t_i и s_i связываются с Entity и Activity модели PROV [16]; модель, конфигурация, конвейер и наборы данных представляются как элементы спецификации вычислительного эксперимента [17]; причинные связи между входными элементами потока и событием r_i фиксируются средствами мелкозернистой provenance [18]; запись j_i включается в проверяемую цепочку аудита с доказательствами существования и целостности [19]. Состояние o_i связывает решение с Agent оператора. При этом слой обмена обязан сохранять $G(r_i)$, $Q(r_i)$, H_i , итоговый статус и машиночитаемую причину отказа. Иначе система сможет описать происхождение, но потеряет правило допуска к реакции. Тем самым существующие модели provenance и проверяемого журналирования отвечают

на вопросы «как сформирован результат» и «сохранилась ли целостность его аудиторского следа», а предлагаемый контракт дополнительно отвечает на вопрос «может ли он инициировать реакцию при действующей эксплуатационной политике».

Практическая значимость состоит в реализации модели как схемы данных, слоя проверок и набора контрактных тестов. Ключевыми элементами являются структура объекта, семантика отказа, идемпотентность, управляемое версионирование и воспроизводимый расчет показателей. Контроль входа, модели и среды соответствует подходам к защите алгоритмов искусственного интеллекта в видеонаблюдении [20].

Ограничения связаны с внутренним стендом, одним аналитическим стеком, одним набором сценариев, 27 камерами, ограниченными операторским экспериментом и отсутствием независимой проверки. Иерархический анализ учитывает наблюдаемую кластеризацию, однако небольшое число верхнеуровневых кластеров ограничивает точность оценки случайных эффектов. В операторской части изменение программного контракта было связано с изменением карточки события, поэтому причинный вклад интерфейса и трассируемости полностью не разделен. При замене компонентов необходимы повторные интеграционная, статистическая и нагрузочная проверки.

Масштабируемость журнала аудита за пределами одного узла отдельно не исследовалась. Средний размер сериализованной записи r_i вместе с j_i составил 1,28 КБ; при нагрузочной интенсивности 512 событий/с это соответствует 56,6 ГБ/сут. и 1,70 ТБ за 30 сут. без учета индексов и резервных копий. Вынесение неизменяемых полей модели, конфигурации и профиля среды в версионизируемые справочники с сохранением в событии ссылок и контрольных сумм уменьшало расчетный объем записи до 0,47 КБ, или 20,8 ГБ/сут. Для промышленной эксплуатации необходимы партиционирование и ротация по времени, хранение горячего журнала в течение ограниченного периода, сжатие закрытых разделов и выгрузка во внешнее неизменяемое или объектное хранилище с периодической проверкой контрольных сумм. Влияние такой схемы на задержку запросов, восстановление и стоимость хранения требует отдельного эксперимента.

Заключение

Разработана программная модель трассируемого контура видеоаналитики, в которой отдельный результат представлен вер-

сионизируемым объектом, связывающим источник, модель, конфигурацию, качество кадра, среду исполнения, архив, аудит и операторское подтверждение. Введено разделение жестких предусловий и мягкой оценки доверия: неполнота происхождения не может быть компенсирована высокой уверенностью нейронной сети. На 360 видеосценариях доля событий с версией модели и конфигурации возросла с 0,44 до 0,98, с архивной ссылкой – с 0,66 до 0,95, с качеством входа – с 0,31 до 0,93. Доля событий, одновременно выполнивших $G(r_i)=1$, увеличилась с 0,230 до 0,866. Медиана времени операторского подтверждения снизилась на 33,6%, а доля отрицательных окон с неverified реакцией – с 0,18 до 0,04. Контрактные тесты показали блокирование неподтвержденных изменений и понижение доверия при неполной архивной связности. Нагрузочная проверка показала увеличение медианы задержки на 0,37 мс и снижение предельной пропускной способности на 8,8% при сохранении более чем шестикратного запаса относительно нагрузки 512 событий/с.

Результаты согласуются с рабочей гипотезой в части трассируемости, управляемости и операционной эффективности результата в пределах испытанной конфигурации. При неизменном аналитическом стеке различия в полноте происхождения и доле неverified реакций относятся к программному контракту, тогда как временной эффект характеризует интегрированный профиль вместе с интерфейсом карточки события. Перенос на другой стек требует интеграционной и нагрузочной проверки.

Модель может служить основой интерфейсов, схемы данных и программы испытаний видеоаналитики, отделяя качество распознавания от доказуемости происхождения и корректности программного жизненного цикла результата.

Список литературы

1. Кобринский Б. А. Доверие к технологиям искусственного интеллекта // Искусственный интеллект и принятие решений. 2024. № 3. С. 3-17. URL: <https://www.mathnet.ru/rus/iipr594> (дата обращения: 10.07.2026). DOI: 10.14357/20718594240301. EDN: HNRFBG.
2. Авдошин С. М., Песоцкая Е. Ю. Доверенный искусственный интеллект как способ цифровой защиты // Бизнес-информатика. 2022. Т. 16. № 2. С. 62-73. URL: <https://bijournal.hse.ru/article/view/25952> (дата обращения: 10.07.2026). DOI: 10.17323/2587-814X.2022.2.62.73.
3. Kreuzberger D., Kühl N., Hirschl S. Machine Learning Operations (MLOps): Overview, Definition, and Architecture // IEEE Access. 2023. Vol. 11. P. 31866-31879. DOI: 10.1109/ACCESS.2023.3262138.
4. Serban A., van der Blom K., Hoos H., Visser J. Software engineering practices for machine learning – Adoption, effects,

- and team assessment // *Journal of Systems and Software*. 2024. Vol. 209. Article 111907. DOI: 10.1016/j.jss.2023.111907.
5. Amershi S., Begel A., Bird C., DeLine R., Gall H., Kamar E., Nagappan N., Nushi B., Zimmermann T. Software Engineering for Machine Learning: A Case Study // 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice. 2019. P. 291-300. DOI: 10.1109/ICSE-SEIP.2019.00042.
 6. Breck E., Cai S., Nielsen E., Salib M., Sculley D. The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction // 2017 IEEE International Conference on Big Data. 2017. P. 1123-1132. DOI: 10.1109/BigData.2017.8258038.
 7. Mitchell M., Wu S., Zaldivar A., Barnes P., Vasserman L., Hutchinson B., Spitzer E., Raji I. D., Gebru T. Model Cards for Model Reporting // *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 2019. P. 220-229. DOI: 10.1145/3287560.3287596.
 8. Myllyaho L., Raatikainen M., Männistö T., Nurminen J. K., Mikkonen T. On misbehaviour and fault tolerance in machine learning systems // *Journal of Systems and Software*. 2022. Vol. 183. Article 111096. DOI: 10.1016/j.jss.2021.111096.
 9. Aliev M. A., Kunina I. A., Kazbekov A. V., Arlazarov V. L. Algorithm for choosing the best frame in a video stream in the task of identity document recognition // *Computer Optics*. 2021. Vol. 45. № 1. P. 101-109. URL: <https://computeroptics.ru/eng/KO/Annot/KO45-1/450112e.html> (дата обращения: 10.07.2026). DOI: 10.18287/2412-6179-CO-811.
 10. Chernov T. S., Razumnuy N. P., Kozharinov A. S., Nikolaev D. P., Arlazarov V. V. Image quality assessment for video stream recognition systems // *Tenth International Conference on Machine Vision (ICMV 2017): Proceedings of SPIE*. 2018. Vol. 10696. Article 106961U. P. 1-8. DOI: 10.1117/12.2309628.
 11. Andriyanov N. A., Dementiev V. E., Tashlinskiy A. G. Detection of objects in the images: from likelihood relationships towards scalable and efficient neural networks // *Computer Optics*. 2022. Vol. 46. № 1. P. 139-159. URL: <https://computeroptics.ru/eng/KO/Annot/KO46-1/460117e.html> (дата обращения: 10.07.2026). DOI: 10.18287/2412-6179-CO-922.
 12. Dodge S., Karam L. J. Understanding How Image Quality Affects Deep Neural Networks // 2016 Eighth International Conference on Quality of Multimedia Experience. 2016. P. 1-6. DOI: 10.1109/QoMEX.2016.7498955.
 13. Sultani W., Chen C., Shah M. Real-World Anomaly Detection in Surveillance Videos // 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2018. P. 6479-6488. DOI: 10.1109/CVPR.2018.00678.
 14. Pang G., Shen C., Cao L., van den Hengel A. Deep Learning for Anomaly Detection: A Review // *ACM Computing Surveys*. 2021. Vol. 54. № 2. Article 38. P. 1-38. DOI: 10.1145/3439950.
 15. Guo C., Pleiss G., Sun Y., Weinberger K. Q. On Calibration of Modern Neural Networks // *Proceedings of the 34th International Conference on Machine Learning*. 2017. Vol. 70. P. 1321-1330. URL: <https://proceedings.mlr.press/v70/guo17a.html> (дата обращения: 10.07.2026).
 16. Moreau L., Groth P., Cheney J., Lebo T., Miles S. The Rationale of PROV // *Journal of Web Semantics*. 2015. Vol. 35. № 4. P. 235-257. DOI: 10.1016/j.websem.2015.04.001.
 17. Хританков А. С., Полежаев В. А., Жуликов Г. А., Гальчик М. С., Климин Н. А., Сахаров К. Е., Минченков В. О., Спирин И. В., Крупнов И. И., Якушева С. Ф., Маратканова А. С., Козырев В. И., Костенецкий П. С., Салех Х. М. Выполнение распределенных вычислительных экспериментов на MLops платформе НИУ ВШЭ // *Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика*. 2025. Т. 14. № 2. С. 42-66. DOI: 10.14529/cmse250203.
 18. Palyvos-Giannas D., Gulisano V., Papatriantafyllou M. GeneaLog: Fine-Grained Data Streaming Provenance in Cyber-Physical Systems // *Parallel Computing*. 2019. Vol. 89. Article 102552. DOI: 10.1016/j.parco.2019.102552.
 19. Putz B., Menges F., Pemul G. A Secure and Auditable Logging Infrastructure Based on a Permissioned Blockchain // *Computers & Security*. 2019. Vol. 87. Article 101602. DOI: 10.1016/j.cose.2019.101602.
 20. Мазин А. В., Белкин Е. Н. Анализ кибератак на алгоритмы искусственного интеллекта в системах видеонаблюдения и методы противодействия // *Известия Института инженерной физики*. 2026. № 1 (79). С. 7-10. URL: https://elibrary.ru/download/elibrary_88923315_47287270.pdf (дата обращения: 10.07.2026).

Конфликт интересов: Авторы заявляют о наличии следующих конфликтов интересов: Автор является сотрудником АО «АЛГОНТ». Организация предоставила испытательную инфраструктуру и видеоданные. Автор самостоятельно выполнил постановку задачи, статистический анализ, интерпретацию результатов и подготовку рукописи. Иные конфликты интересов отсутствуют.

Conflict of interest: The author declares the following conflicts of interest: The author is an employee of ALGONT JSC. The organization provided the testing infrastructure and video data. The author independently formulated the problem, performed the statistical analysis and interpretation of results, and prepared the manuscript. There are no other conflicts of interest.

Финансирование: Исследование выполнено без внешнего грантового финансирования с использованием инфраструктуры АО «АЛГОНТ».

Financing: The study received no external funding, but utilized ALGONT JSC infrastructure.