УДК 004.89

МЕХАНИЗМЫ АУДИТА И ОТСЛЕЖИВАНИЯ В ДОВЕРЕННЫХ ML-ФРЕЙМВОРКАХ

Романов А.В., студент гр. 605, II курс
Московский государственный университет имени М. В. Ломоносова,
г. Москва

По мере того, как модели машинного обучения становятся неотъемлемой частью цифровой инфраструктуры в здравоохранении, промышленности, юриспруденции, финансах и других социально значимых сферах, растёт потребность в обеспечении полной прозрачности их функционирования. Речь идёт не только о возможности интерпретировать предсказания модели, но и о том, чтобы документировать весь жизненный цикл работы системы: от поступления данных до вывода результатов. Именно в этом контексте приобретают принципиальное значение механизмы аудита и отслеживания, встроенные в доверенные ML-фреймворки. Такие механизмы необходимы не только для устранения ошибок и расследования инцидентов, но и для соблюдения нормативных требований, обеспечения цифровой ответственности и укрепления доверия со стороны пользователей, экспертов и регулирующих органов. Настоящая работа посвящена системному анализу архитектурных решений, функциональных возможностей и перспектив развития механизмов аудита и отслеживания в доверенных фреймворках машинного обучения [1].

Под аудитом в машинном обучении понимается формализованное и воспроизводимое отслеживание всех событий, параметров, данных и вычислений, происходящих в рамках системы. Отслеживание (или трассировка) включает в себя сбор и структурирование информации о ходе выполнения алгоритмов, изменениях в состоянии модели, использовании вычислительных ресурсов, а также об ошибках, предупреждениях и отклонениях от ожидаемого поведения. В доверенном ML-фреймворке аудит должен охватывать все этапы: от подготовки данных и настройки гиперпараметров до процессов инференса и повторного обучения модели.

Одной из ключевых задач таких механизмов является обеспечение воспроизводимости и доказуемости результатов. Все действия – будь то запуск модели, изменение конфигурации, трансформация данных или обновление веса – должны быть логгированы и снабжены уникальными идентификаторами. Это позволяет не только повторить эксперимент в том же окружении, но и доказать, что полученные результаты были получены честным

путём, без манипуляций, ошибок или скрытых вмешательств. Современные фреймворки начинают использовать технологии immutable logs (неизменяемые журналы), основанные на хэшировании и цепочках проверок, что повышает стойкость к фальсификации и обеспечивает юридическую доказательность записей [2].

Механизмы отслеживания должны быть тесно интегрированы с системами управления данными и метаданными. Это включает в себя автоматическую регистрацию происхождения данных (data provenance), версий обучающих выборок, а также условий, при которых данные были собраны, преобразованы или анонимизированы. Особенно это важно при работе с персональной информацией, где требуются доказательства соответствия политике обработки данных, например в рамках GDPR, HIPAA или AI Act.

Отдельное внимание следует уделить архитектуре логгирования и системам хранения. В условиях масштабных вычислений и распределённых сред необходима надёжная и масштабируемая инфраструктура для сбора и хранения логов. Многие доверенные ML-фреймворки реализуют распределённое логгирование на основе журналов событий, брокеров сообщений и централизованных хранилищ (например, на базе Elastic Stack, Fluentd, Apache Kafka и пр.). Для эффективной аналитики и ретроспективного анализа требуется также продвинутая система поиска, фильтрации и визуализации логов [3].

Важным направлением является внедрение механизмов активного аудита и автоматического анализа логов. Это позволяет не только фиксировать действия, но и в реальном времени обнаруживать отклонения, подозрительную активность, а также выявлять неявные закономерности в поведении системы. Такие функции могут реализовываться на основе правил (rule-based systems), а также с помощью специализированных моделей машинного обучения, обученных на журналируемых событиях. В случае аномалий система может автоматически инициировать оповещение, запуск аварийного протокола или приостановку работы модели.

Особо стоит выделить вопрос доверия к самому механизму аудита. В доверенных системах недостаточно просто фиксировать информацию – необходимо гарантировать её достоверность, целостность и неизменяемость. Это требует применения криптографических методов (например, цифровых подписей, Merkle-деревьев, блокчейн-подобных структур) и создания процедур верификации логов независимыми сторонами. В перспективе это создаёт возможность формирования юридически значимых доказательств корректности работы системы, что особенно важно в правовых или медицинских сценариях.

Дополнительный вызов представляет собой аудит моделей, обучающихся в реальном времени или с использованием потоков данных. В таких случаях стандартные методы логгирования и верификации могут оказаться неэффективными из-за объёма и скорости информации. Для таких сценариев разрабатываются облегчённые, иерархические и событийно-триггерные модели аудита, способные фиксировать ключевые этапы без избыточной нагрузки на систему [4].

С точки зрения нормативного соответствия, механизмы аудита в доверенных фреймворках должны поддерживать генерацию отчётов, пригодных для предоставления регуляторам, аудиторам и заинтересованным сторонам. Такие отчёты должны включать информацию о жизненном цикле моделей, изменениях конфигураций, идентификации операторов, участвовавших в обучении или запуске, а также об обработанных данных и условиях принятия решений.

В заключение, механизмы аудита и отслеживания являются краеугольным камнем архитектуры доверенных ML-фреймворков. Их роль выходит за рамки обеспечения технической корректности – они становятся инструментом правовой и этической ответственности, основой для прозрачности, доверия и сертификации. Развитие этой области требует тесного сотрудничества между разработчиками, специалистами по безопасности, регуляторами и представителями общества. В будущем ключевыми направлениями станут автоматизация анализа аудиторских данных, стандартизация форматов логгирования, использование защищённых аппаратных сред и формальная верификация трассировок, что позволит создать полноценную инфраструктуру доверенного и подотчётного машинного обучения [5].

Список литературы:

1. Amershi, S. Software Engineering for Machine Learning: A Case Study / S. Amershi, A. Begel, C. Bird [et al.] // IEEE/ACM International Conference on Software Engineering. – 2019. – P. 291–300. – DOI 10.1109/ICSE-SEIP.2019.00042.

2. Schelter, S. Automatically Tracking Metadata and Provenance of Machine Learning Experiments / S. Schelter, F. Biessmann, T. Januschowski [et al.] // Machine Learning Systems Workshop at NeurIPS. – 2017. – P. 1–7.

3. Renggli, C. Continuous Integration of Machine Learning Models with Ease.ml/ci / C. Renggli, L. Rimanic, N. M. Gürel [et al.] // Proceedings of Machine Learning and Systems. – 2019. – Vol. 1. – P. 341–352.

4. Sculley, D. Machine Learning: The High Interest Credit Card of Technical Debt / D. Sculley, G. Holt, D. Golovin [et al.] // SE4ML: Software Engineering for Machine Learning. – 2014. – P. 1–9.

5. Polyzotis, N. Data Management Challenges in Production Machine Learning / N. Polyzotis, S. Roy, S. E. Whang, M. Zinkevich // Proceedings of the ACM SIGMOD International Conference on Management of Data. – 2017. – P. 1723–1726. – DOI 10.1145/3035918.3054782.