

Модуль: Коридор нормальности (Normality Corridor Module)

Тип

Аналитико-оценочный модуль институционального допуска

Назначение

Непрерывно анализирует поведенческие траектории участников (водителей, операторов, парков) и формирует динамическую модель нормы — основанную не на фиксированных порогах, а на эмпирических траекториях успешного поведения. Обеспечивает допуск, ограничение, рекомендации и репутационные метки по справедливой, проверяемой логике.

Структура подсистем

Подсистема	Функция	Ключевые данные
trajectory_collector	Извлечение последовательностей действий водителей и операторов из ERP	действия, события, KPI, финансы
group_norm_bounds	Определение границ нормы на основе реер-групп	кластеризация, скользящие границы
normality_score	Композитная метрика отклонения от нормы	Z-score, Mahalanobis, Rank position
flagging_engine	Пометка аномалий (для FSM, Coach, Access)	флаг, сила отклонения, контекст
risk_path_typing	Обнаружение траекторий, ведущих к рискам	просрочки, штрафы, ДТП, убытки
operator_analysis	Оценка операторов: загрузка, KPI, поведенческие траектории	idle_rate, загрузка, замена водителей
response_context	Передача отклонений в Coach, Access и IMS	полное описание паттерна и реакции
meta_evaluation	Проверка справедливости и устойчивости самих норм	bias-detect, fairness audit, SDR-отчёты

Алгоритмы и методы

Метод	Назначение
Z-score / Mahalanobis	Выявление отклонений от нормы
Peer Group Ranking	Позиция участника относительно когорты
Sequence Classification	Типизация поведенческих траекторий
Segment Filtering	Учёт региона, ТС, роли, сезона
QI-sat (Quartile Intelligent Categorization)	Устойчивое квартильное разбиение при малых/смещённых выборках
Trajectory Early Warning	Раннее выявление схожести с паттернами прошлых инцидентов (штрафы, ДТП)

Метод	Назначение
Causal Inference (в перспективе)	Связь между действиями и последствиями (ДТП, штрафы и т.д.)

Источники данных

Источник	Обновление
ERP Feature Store batch / near real-time	
event_log	в реальном времени
telemetry_engine	потокowo / 15 сек
peer_groups	пересчёт при накоплении новых траекторий
contract_status	по завершению аренды или расторжению
meta_audit_data	ежедневно / еженедельно

Режим работы и поток данных

- Извлечение и сбор**
 - ERP передаёт батчи фич и событий
 - Сбор слайсов траекторий по ID субъекта
- Обработка и нормализация**
 - Группировка по peer-группам
 - Расчёт отклонений и норм
- Вывод**
 - Метки: normal, risky, anomalous, compliant
 - Передача в Coach, FSM, IMS, Access Layer
 - Логгирование в аудит и feature store

Пример вывода (Feature Store → Consumers)

Поле	Описание	Обновление
driver_normality_score	Оценка нормальности по композитной модели	batch
operator_efficiency_z	Z-оценка KPI оператора	batch
anomaly_flag	0 / 1 (аномалия)	real-time
risk_path_type	Класс опасной траектории	batch
meta_bias_score	Метрика возможного перекоса нормы	еженедельно

API-сценарии

Версия: v1.0 (backward-compatible)
 Авторизация: OAuth2 (JWT Bearer)

- Получить нормальность участника
GET /api/v1/norm/actor/{id} → {score, flag, peers}

2. Проверить норму группы

GET /api/v1/norm/group/{segment} → {mean, σ , bounds}

3. Обновить реер-группу

POST /api/v1/norm/peer_group/{group_id}/refresh

4. Получить отклонения по оператору

GET /api/v1/norm/operator/{operator_id}

5. Аудит справедливости нормы

GET /api/v1/norm/meta/bias_audit

SLA, производительность и батчи

- Обновление норм: ≤ 10 мин после батча
 - Импорт фич: ≤ 2 мин / 1000 записей
 - Задержка API: ≤ 250 мс (p95)
 - Поддержка ≥ 5000 участников в ежедневной оценке
 - Масштабируемость до 100k участников — через BQ + сегментную обработку
-

Безопасность и защита данных

- Аутентификация: OAuth2 (JWT Bearer)
 - Разграничение доступа: RBAC
 - Шифрование: TLS 1.2+ (в пути), AES-256 (в покое)
 - Логгирование всех аномалий и решений → immutable audit trail
 - РП и поведенческие данные — под защитой GDPR
 - "Right to explanation" — метки объяснимы (SHAP, LIME)
-

Наблюдаемость и CI/CD

- Логи: Fluentd → ELK
 - Метрики: Prometheus + Grafana
 - CI/CD: GitHub Actions → Docker → Deploy
 - ETL: cron + Airflow DAGs
 - Bias-Check: еженедельно → DataHub lineage + alert если перекос $> 15\%$
-

Соответствие и аудит

- SHAP/LIME: объяснимость выводов
 - Basel III: интерпретируемость, отсутствие black-box
 - ESG / SDG: нечувствительность к полу, нации, возрасту
 - Fairness Audit: скан на избыточные отклонения в нормах
 - Data Lineage: <https://datahub.tf/norm-corr-lineage>
-

✂ Резерв и восстановление

- Бэкапы: раз в сутки + дифференциально каждые 6 ч
 - RTO: ≤ 1 ч
 - RPO: ≤ 24 ч
 - DR-тесты: раз в квартал
-

🔑 Ключевые кейсы

1. 🚩 **Ограничение водителя**
Если `driver_normality_score < -2.5` → `FSM-token = restricted` → коуч
2. 🚧 **Допуск к дорогим активам**
Если `score > 0.5` и `no flags` → доступ к автомобилю класса B+
3. 🧠 **Коучинг-совет**
Стиль сна, тип штрафов, частота обращений → рекомендация по ротации
4. ⚠ **Оператор с неэффективным парком**
Высокий `idle_rate` + перераспределения → предупреждение, оценка репутации

Простыми словами:

Представь, что ты управляешь большой системой аренды автомобилей: у тебя сотни машин, десятки водителей, и всё это крутится каждый день. Кто-то аккуратен, кто-то безалаберный. Кто-то доводит машину до убытков, а кто-то бережёт. Как заранее понять, кто приведёт к проблемам, а кто — нет? Как вовремя заметить, что кто-то **начал "съезжать с трассы"**?

Вот тут и нужен модуль **"Коридор нормальности"**.

🔑 Зачем нужен

Он отслеживает, как ведут себя участники (водители и операторы), и **сравнивает их поведение с тысячами других**. Если кто-то **повторяет траекторию тех, кто в прошлом попал в ДТП, получил кучу штрафов или угробил машину**, система это заметит — ещё до того, как случится беда.

Это не просто "проверка по списку", а **умная, обучающаяся система**, которая говорит:

"Вот у тебя водитель — он вроде нормальный, но пошёл по дорожке, которая у 82% других водителей заканчивалась проблемами. Предупреждаю заранее."

🧠 Как работает

1. **Смотрит на поведение всех** — кто, как водит, как часто попадает в штрафы, насколько загружен автопарк, как часто машины простаивают, кто с кем работает и что из этого выходит.
 2. **Запоминает успешные траектории** — те, что привели к хорошим результатам: прибыль, отсутствие штрафов, сохранность машин.
 3. **Запоминает провальные траектории** — где были проблемы: убытки, поломки, ДТП.
 4. **Сравнивает нового участника или текущее поведение** с этими шаблонами. Если находит сходство с плохими — бьёт тревогу. Если всё ок — даёт зелёный свет.
-

Что даёт

- **Предупреждение заранее** — можно отключить проблемного водителя до того, как он устроит ДТП.
 - **Честный и проверяемый допуск** — не по субъективному мнению, а по статистике.
 - **Оценка операторов** — видно, кто управляет парком эффективно, а кто — не умеет.
 - **Формирует доверие у инвесторов** — система показывает, что в ней не может случайно затесаться разрушительный участник.
 - **Снижение издержек** — меньше аварий, штрафов, убытков.
-

Что меняет

До него:

- Работали "по наитию", надеясь на опыт менеджера.
- Поздно замечали, что кто-то ведёт к проблеме.
- Не было прозрачного критерия — кого пускать, а кого нет.

С ним:

- **Решения основаны на данных.**
 - **Каждый шаг водителя и оператора — под наблюдением.**
 - **Нормы не заданы жёстко — они живые, адаптируются к реальности.**
 - **Даже если нет большого объёма данных — работает QI-метод, который умеет оценивать устойчивость по квартилям, а не по средним, что даёт хорошую точность даже на малых выборках.**
-

Пример:

Водитель Иван недавно начал работать. Он пунктуален, но слишком агрессивно ускоряется и часто едет выше разрешённой скорости. Система видит, что похожее поведение у других водителей заканчивалось 60%

аварийностью и кучей штрафов.

Рекомендация: ограничить допуск, направить на коучинг.

🔒 И всё это — **с объяснением**. То есть можно открыть отчёт и понять: *почему система посчитала этого водителя отклонением?* Это важно для доверия и для аудиторов.