📘 Module: Normality Corridor Module

Type
Analytical & evaluation module for institutional-grade access control

---

🎯 Purpose

Continuously analyzes behavioral trajectories of participants (drivers, operators, fleets) and forms a dynamic model of "normality"—based not on fixed thresholds, but on empirical patterns of successful behavior. Enables access, restrictions, recommendations, and reputation tags through a fair and verifiable logic.

🧩 Subsystem Structure

Subsystem	Function	Key Data
trajectory_collector	Extracts sequences of driver/operator actions from ERP	actions, events, KPIs, financials
group_norm_bounds	Defines normality bounds based on peer groups	clustering, sliding boundaries
normality_score	Composite score for deviation from normal	Z-score, Mahalanobis, rank position
flagging_engine	Flags anomalies (for FSM, Coach, Access)	flag, deviation strength, context
risk_path_typing	Detects risk-prone behavior trajectories	delinquencies, fines, accidents, losses
operator_analysis	Assesses operator load, KPIs, and behavioral trends	idle_rate, load, driver turnover
response_context	Transmits deviation context to Coach, Access, IMS	pattern descriptors and triggers
meta_evaluation	Audits fairness and robustness of normality logic	bias detection, fairness audit, SDRs

---

🔍 Algorithms & Methods

Method	Purpose
Z-score / Mahalanobis	Detect statistical deviation from norm
Peer Group Ranking	Assess participant position within cohort
Sequence Classification	Identify behavioral trajectory types
Segment Filtering	Contextual filtering (region, vehicle type, role, season)
QI-sat (Quartile Intelligence)	Robust quartile-based scoring for small or skewed datasets
Trajectory Early Warning	Detect similarity to past incident patterns (accidents, fines)
Causal Inference (future)	Identify causality between actions and outcomes

---

📦 Data Sources

Source	Frequency
ERP Feature Store	batch / near real-time
event_log	real-time
telemetry_engine	streaming / every 15 sec
peer_groups	upon accumulation of new trajectories
contract_status	upon rental end or termination
meta_audit_data	daily / weekly

🔁 Operational Mode & Data Flow

1. **Extraction & Ingestion**
   - ERP sends feature/event batches
   - Trajectories sliced by subject ID
2. **Processing & Normalization**
   - Grouping into peer cohorts
   - Deviation and normality scoring
3. **Output**
   - Tags: normal, risky, anomalous, compliant
   - Sent to Coach, FSM, IMS, Access Layer
   - Logged into audit and feature store

🧪 Feature Store Output Example (→ Consumers)

Field	Description	Update Frequency
driver_normality_score	Composite normality score	batch
operator_efficiency_z	Z-score of operator KPIs	batch
anomaly_flag	0 / 1 anomaly flag	real-time
risk_path_type	Risk trajectory classification	batch
meta_bias_score	Potential bias metric in peer grouping	weekly

---

🔌 API Scenarios

Version: v1.0 (backward-compatible)
Authorization: OAuth2 (JWT Bearer)

1. Get actor normality score
   `GET /api/v1/norm/actor/{id}` → `{score, flag, peers}`
2. Check group normal bounds
   `GET /api/v1/norm/group/{segment}` → `{mean, σ, bounds}`
3. Refresh peer group
   `POST /api/v1/norm/peer_group/{group_id}/refresh`
4. Get operator deviation report
   `GET /api/v1/norm/operator/{operator_id}`

5. Audit fairness of normality logic
`GET /api/v1/norm/meta/bias_audit`

⏱️ SLA, Performance & Batching

- Norm updates: ≤ 10 min post-batch
- Feature import: ≤ 2 min / 1,000 records
- API latency: ≤ 250 ms (p95)
- Supports ≥ 5,000 daily-evaluated participants
- Scalable to 100k via BQ + segmented processing

🛡️ Security & Data Protection

- Authentication: OAuth2 (JWT Bearer)
- Authorization: RBAC
- Encryption: TLS 1.2+ (in transit), AES-256 (at rest)
- All anomaly decisions are logged in immutable audit trail
- PII & behavioral data are GDPR-compliant
- "Right to explanation" supported (SHAP, LIME)

🔭 Observability & CI/CD

- Logs: Fluentd → ELK
- Metrics: Prometheus + Grafana
- CI/CD: GitHub Actions → Docker → Deploy
- ETL: cron + Airflow DAGs
- Bias check: weekly → DataHub lineage + alert if skew > 15%

✅ Compliance & Auditing

- SHAP / LIME: Explainable outputs
- Basel III: Interpretable decisions, no black-box
- ESG / SDG: No sensitivity to gender, race, age
- Fairness Audit: Scan for norm bias
- Data Lineage: https://datahub.tf/norm-corr-lineage

🛠️ Backup & Recovery

- Daily backups + 6h incremental diffs
- RTO: ≤ 1h
- RPO: ≤ 24h

- DR drills: quarterly

📌 Key Use Cases

1. 🚩 Driver Restriction
   If `driver_normality_score < -2.5` → FSM-token = restricted → send to Coach
2. 🏁 Access to premium assets
   If `score > 0.5` and no flags → eligible for B+ vehicle class
3. 🧠 Coaching Recommendation
   Sleep style, fine types, contact frequency → suggest driver rotation
4. ⚠️ Inefficient operator detection
   High idle_rate + driver churn → warning + reputation downgrade

---

🧠 Explaining It in Plain Language

Imagine you're running a large vehicle rental operation — hundreds of vehicles, dozens of drivers. Some are careful, others reckless. Some drivers quietly destroy profit margins. But how do you **know in advance** who will cause problems? How can you spot that someone is **veering off the safe path**?

This is where the **Normality Corridor Module** comes in.

📌 Why It Matters

It tracks how participants behave and compares them to thousands of historical cases.
If a person starts repeating the patterns of others who ended up with crashes or fines, the system will detect it **before it's too late**.

It's not just a checklist. It's an intelligent system that says:

"This driver *looks fine*, but their behavior is **tracking** the same path that led 82% of other drivers into serious trouble."

🧠 How It Works

1. Observes everyone — who drives how, how often fines occur, vehicle downtime, operator performance.
2. Remembers **successful trajectories** — those that led to profit, no fines, well-kept vehicles.
3. Remembers **failure trajectories** — those that led to damage, debt, or accidents.
4. Compares new or current actors to these patterns.
   If there's a match with risky paths → **raises the alarm**.
   If everything is fine → gives a green light.

✅ What It Brings

- **Early warning** — identify drivers on dangerous paths before a crash happens.
- **Fair and explainable access control** — no more "gut feeling" or bias.
- **Operator assessment** — see who runs their fleet efficiently.
- **Investor trust** — strong risk screening builds credibility.
- **Lower costs** — fewer accidents, fewer fines, less financial loss.

🔄 What Changes

Before:

- Relied on intuition, anecdotal experience.
- Spotted risk **too late**.
- No clear rules on who gets in or out.

With the Module:

- **Data-driven decisions**
- Every actor's behavior is continuously monitored.
- Norms are **adaptive**, not hard-coded.
- Even with limited data, the **QI method** enables robust analysis using quartile logic — giving precision where averages fail.

💬 **Example:**
Driver Ivan just joined. He's punctual but accelerates too hard and often exceeds speed limits. The system notices this matches past patterns that led to 60% accident rate and heavy fines.
Recommendation: restrict access, send to coaching.

🔒 And all this is **explainable**. Reports show **why** a person is flagged. This is crucial for trust and regulatory audits.