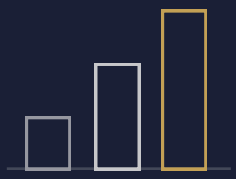
THE ELECTE QUARTERLY

NO. 2 — NOVEMBER 2026



On distinguishing meaningful signals from noise

ELECTE.

Contents

THE QUARTERLY ESSAY

When Data Breaks the Pattern

Machine learning and the science of anomaly detection

01 Machine Learning for Anomaly Detection 6

02 Core Algorithms and Approaches 7

03 Data Preparation and Feature Engineering 9

04 Model Evaluation and Pitfalls 11

05 Business Use Cases 12

06 Autonomous Analytics Workflows 13

07 Key Takeaways For Your Team 14

When Patterns Begin to Shift

Uncovering anomalies in time series

01 Spotting the Signals 17

02 Anomalies in Time Series 18

03 Statistical, ML, and AI Detection Methods 19

04 Detection Performance with the Right Metrics 20

05 Batch and Streaming Anomaly Monitoring 22

06 Business Scenarios 23

07 Choosing the Right Tools 24

08 Practical Best Practices 25

Teaching AI to Spot the Unexpected

AI-powered anomaly detection for business data

Contents

THE QUARTERLY ESSAY

01	Why Anomaly Detection Matters	29
02	What Counts as an Anomaly	30
03	How Anomaly Detection Algorithms Work	32
04	Choosing the Right Evaluation Metric	34
05	Real-World Use Cases Across Industries	35
06	Common Pitfalls	37
07	Tuning Best Practices	38
08	The Essentials	40

When Data Breaks the Pattern

Machine learning and the science of anomaly detection

Behind every anomaly lies a departure from what we expect. Understanding these deviations is where machine learning turns raw data into a deeper picture of business behavior.

BY FABIO LAURIA — CEO & FOUNDER, ELECTE

You can have a dashboard that looks healthy, a weekly report that feels reassuring, and still miss the one signal that matters. A churn spike can start as a tiny shift in behavior, an inventory problem can hide inside “normal” variance, and a fraud pattern can sit just outside the thresholds your team checks by hand.

That's where machine learning for anomaly detection earns its place, it finds the rare events that don't fit the pattern, especially when the business is too busy for people to watch every stream all day.

For business leaders, this isn't about clever math for its own sake. It's about catching problems early enough to protect margin, reduce waste, and keep operations moving before a small deviation turns into a customer-facing issue. For analysts, it's a practical way to move from passive reporting to active monitoring, where the model watches for what shouldn't be happening right now.

01 Machine Learning for Anomaly Detection

Not every deviation is accidental, and not every anomaly is obvious

A retailer can stare at a clean dashboard and still miss a subtle drop in inventory turnover, just like a finance team can miss a slow fraud pattern that doesn't break a hard rule. Machine learning for anomaly detection is the capability that spots those rare items, events, or observations that differ from the rest of the data enough to raise suspicion. It's less like reading a monthly report and more like having a watchful analyst who notices when the story changes midstream.

That idea has a long history. A 2024 review traces general anomaly detection thinking back to 1777, when Bernoulli's work addressed how to accept or reject extreme observations, while the first time-series-specific work appeared in 1957 and Fox's 1972 study was among the first to define anomalous behavior across time, and the same review says 65% of methods published between 1980 and 2000 were unsupervised, showing how early the field leaned toward learning normal patterns without labels.[1]

1 Dive into Time-Series Anomaly Detection: A Decade Review, Paul Boniol, 29 Dec 2024

BI Explains the Past; Anomaly Detection Flags the Present

Standard business intelligence usually answers questions like “What was revenue last week?” or “Which channel converted best?”. That's useful, but it's backward-looking. Anomaly detection is different because it watches for deviations while the data is still moving, which is why it's so valuable in environments where delays cost money or trust.

FIG. 1 UNDERSTANDING MACHINE LEARNING



A practical way to think about it is this:

ROLE	FUNCTION	PURPOSE
Dashboards	*Summarize*	*Reveal trends and past performance*
Anomaly models	*Monitor*	*Flag deviations from expected patterns*
Operational teams	*Act*	*Investigate alerts before issues spread*

If a metric matters every hour, not just every month, you need anomaly detection thinking, not just reporting.

02 Core Algorithms and Approaches

Different methods reveal different kinds of irregularities, each with its own strengths and limitations

The easiest way to choose an anomaly method is to start with your data reality, not the algorithm name. If you have labeled incidents, you can teach a model what bad looks like. If you don't, you need methods that learn normal behavior first and treat deviation as a warning sign.

The Four Main Approaches

Statistical methods compare each value to a rule or threshold. They're simple, quick to explain, and often a useful starting point when teams want immediate visibility. Supervised methods use labeled examples of normal and abnormal events, which can work well when you already know what failure looks like.

Semi-supervised methods learn mostly from normal data and then judge new points against that baseline. They're a strong middle ground when incidents are rare and labels are incomplete.

Unsupervised methods look for structure in the data itself, which makes them appealing when you have lots of events but few confirmed anomalies.

Choosing Algorithms for Business Needs

Isolation Forests are often practical for SMEs because they isolate unusual points rather than trying to model every normal pattern in detail. Autoencoders learn compressed representations of normal data and struggle to reconstruct unusual records, which makes them useful when patterns are dense and repeatable. One-Class SVMs can draw a boundary around what "normal" looks like, while clustering methods and probabilistic models help when your data naturally groups into several operating modes.

The best fit depends on data maturity, not vendor hype. If your team has little incident history, unsupervised approaches are often the most realistic place to start. If you have a stable annotation process, supervised or semi-supervised approaches can improve precision, especially in high-stakes workflows.

A broad benchmark study evaluated 30 algorithms across 57 datasets and ran 98,436 experiments, and its core message was clear, algorithm choice should depend on supervision level and anomaly type rather than a single winner.[2]

2 ADBench: Anomaly Detection Benchmark, Songqiao Han, 17 September 2022

DETECTION TYPE	DATA REQUIREMENT	KEY ALGORITHMS	BEST BUSINESS USE CASE
Statistical	*Minimal history, clear thresholds*	*Z-score, IQR, moving baselines*	*Simple monitoring and quick alerts*
Supervised	*Labeled normal and anomalous cases*	*Logistic regression, tree models, neural nets*	*Known fraud, known failures, known incidents*
Semi-supervised	*Mostly normal data, few anomaly labels*	*One-Class SVM, autoencoders*	*Rare incident detection with limited labels*
Unsupervised	*Unlabeled or weakly labeled data*	*Isolation Forest, clustering, probabilistic models*	*SMEs starting from raw event streams*

You don't pick the "best" anomaly algorithm in a vacuum, you pick the one your data can actually support.

03 Data Preparation and Feature Engineering

The quality of an anomaly detection model begins long before the algorithm is chosen

Most anomaly projects fail before modeling starts because the data is messy in ways the dashboard never shows. Missing values, inconsistent units, and unhelpful raw timestamps can make ordinary behavior look suspicious. If one metric is scaled in thousands and another in fractions, the model can overreact to the bigger number and ignore the subtler signal.

Clean the Signal Before You Teach the Model

Start by removing obvious duplicates, fixing timestamp issues, and deciding how to handle gaps. Then normalize or encode values so the model compares like with like. Anomaly detection is sensitive to context, and a dirty input can create false alarms that look clever but don't help anyone act faster.

For time-series and transactional data, features matter as much as rows. Rolling averages help smooth noisy spikes, lag features show what changed from one period to the next, and seasonality indicators tell the model that a Friday surge may be normal in retail but suspicious in finance. When a business has many variables, dimensionality reduction can help shrink the noise without losing the core pattern.

Features That Explain Behavior

A useful feature set often answers a simple question: "What changed relative to the recent past?". That's why ratios, deltas, and moving windows tend to outperform raw values in operational settings. They make the model better at separating a true anomaly from a predictable seasonal spike.

A quick checklist helps keep the work grounded:

- **Audit the source fields:** verify that timestamps, IDs, and event types are consistent.
- **Handle missing values deliberately:** don't let silent gaps become fake anomalies.
- **Create context features:** add rolling windows, lagged values, and seasonality markers.
- **Validate distributions:** make sure one field isn't dominating only because of scale.
- **Keep labels separate:** if you have them, preserve them for evaluation, not feature leakage.

FIG. 2 THE DATA PREPARATION PIPELINE



04 Model Evaluation and Pitfalls

Accuracy alone rarely tells the whole story when the cost of missing a signal can be significant

A model can look excellent on paper and still fail in production if the test setup is unrealistic. That happens a lot in anomaly detection because the data is usually imbalanced, the labels are incomplete, and the definition of "normal" changes over time. In that environment, plain accuracy can be misleading, because a model can be "right" most of the time and still miss the rare events that matter most.

Beyond Accuracy

A recent survey on the practical side of anomaly detection says common datasets remain highly imbalanced, often with too few annotated anomalies for self-supervised or semi-supervised learning, and it notes that performance can collapse under realistic anomaly rates such as 0.1%, sometimes producing zero recall on million-scale graphs.[3] That's a reminder that evaluation must look like production, not a classroom exercise.

3 Generative anomaly detection: a comprehensive review of modeling principles, advances, and future opportunities, Jiaqi Zhu, 1 June 2026

Common Pitfalls

Concept drift is one of the biggest risks. Normal behavior changes as promotions, customer habits, staffing, and system load change, so a model that learned last quarter's baseline can become stale. Alert fatigue is the other major risk, because too many false positives train teams to ignore the system altogether.

A good validation setup should mirror the operating rhythm of the business, not just the structure of the dataset. For multivariate time-series work, mTSBench aggregates 344 labeled time series across 19 datasets, which underscores how dataset-dependent real-world performance can be.[4] That's why a model should always be checked against domain-specific seasonality, event frequency, and label sparsity before anyone trusts it in production.

4 mTSBench: Benchmarking Multivariate Time Series Anomaly Detection and Model Selection at Scale, Xiaona Zhou, February 2026

WHAT TO CHECK	WHY IT MATTERS
Precision and recall	*Shows whether alerts are useful and complete*
F1-score	*Balances missed anomalies and false alarms*
Time-based validation	*Tests whether the model survives changing conditions*
Domain-specific slices	*Reveals whether the model fails on certain products, regions, or channels*

05 Business Use Cases

From financial irregularities to operational disruptions, anomalies often reveal what routine reporting misses

Anomaly detection becomes easier to justify when you tie it to a cost center or a risk bucket. In finance, the obvious use case is fraud and AML monitoring, where the value lies in catching suspicious patterns quickly enough to reduce exposure and route cases to the right reviewers. In retail, the payoff is inventory and promotion monitoring, especially when stock depletion or discount behavior doesn't match the usual sales pattern. In operations, it supports predictive maintenance and logistics monitoring by flagging process changes before they become downtime or delays.

Where the Data Usually Comes From

Finance teams often work from transactions, account activity, and entity relationships. Retail teams monitor SKU movement, basket behavior, pricing, and promo calendars. Operations teams lean on sensor data, maintenance logs, routing events, and service-level metrics.

The business outcome is not the alert itself, it's the decision that follows the alert. A suspicious transaction can be routed faster, a fast-moving SKU can be replenished sooner, and a route deviation can be reviewed before it hits service levels. That's why anomaly detection matters most when it's attached to a clear response process.

Why Autonomous Monitoring Matters

Many teams know they need continuous monitoring, but they don't have the bandwidth to stare at every dashboard. That's where autonomous agents become relevant, because they can watch streams, summarize changes, and hand people only the signals that deserve action.

Operational value comes from reducing review time, not just improving model scores.

06 Autonomous Analytics Workflows

Intelligent agents turn continuous monitoring into actionable insights

Building a model is only half the job. The harder part is keeping it current, watching for drift, and making sure the right person sees the right alert at the right time. That's the “last mile” problem in anomaly detection, and it's where many SMEs get stuck, because manual review doesn't scale with the volume of signals.

Building a model is only half the job. The harder part is keeping it current, watching for drift, and making sure the right person sees the right alert at the right time. That's the “last mile” problem in anomaly detection, and it's where many SMEs get stuck, because manual review doesn't scale with the volume of signals.

From Model Maintenance to Continuous Monitoring

An AI-powered data analytics platform can automate the repetitive parts of the workflow, from preprocessing to continuous monitoring. That means less time spent stitching together scripts and dashboards, and more time spent interpreting the patterns that affect revenue or risk.

The key shift is organizational: autonomous systems act as dedicated analysts, monitoring business data, flagging anomalies, and generating reports without manual intervention.

Why It Matters for SMEs

SMEs rarely need more complexity. They need fewer moving parts, clearer alerts, and a path from detection to decision that doesn't require a full data science function. That's what makes autonomous analytics useful, it reduces the gap between “the model found something” and “someone acted on it”.

07 Key Takeaways For Your Team

Effective anomaly detection depends less on finding the perfect model than on asking the right questions of your data

Machine learning for anomaly detection works best when you treat it as an operating capability, not a one-off experiment. Start with the business signal you want to protect, then choose a method that matches your data maturity and alerting needs. If your team is early in the journey, prioritize clean inputs, a sensible baseline, and a review process that prevents alert fatigue.

A practical rollout usually looks like this:

1. **Audit your data streams.** Identify the metrics that matter most and check whether they're complete, timely, and consistent.
2. **Pick the right detection style.** Use labeled methods only when labels are trustworthy, otherwise begin with unsupervised or semi-supervised approaches.
3. **Validate against real operating patterns.** Test on seasonal shifts, sparse anomalies, and the same kinds of drift you see in production.
4. **Attach an action owner.** Every meaningful alert should land with someone who can investigate and respond.
5. **Automate the last mile.** Use a platform or agent layer to monitor, route, and summarize signals continuously.

When Patterns Begin to Shift

Uncovering anomalies in time series

Time rarely moves in straight lines. Recognizing the difference between natural fluctuations and meaningful changes can reveal what conventional reporting overlooks.

BY FABIO LAURIA — CEO & FOUNDER, ELECTE

You can have a dashboard that looks healthy and still miss the problem that matters. A sales dip hides inside normal seasonality, support tickets creep up after a release, or inventory goes missing because an upstream feed stalled, not because demand changed. That's where anomaly detection in time series becomes useful, it turns noisy movement into a clear early warning system for business teams that need to act before customers notice.

The challenge isn't just spotting something unusual. It's deciding whether the alert is real, whether the metric is trustworthy, and whether the signal is actionable for your team. That trust gap is where many programs fail, because a model can look strong on paper and still create confusion in operations. Value comes when your detection method, your evaluation metric, and your data quality checks all line up with the business question you're trying to answer.

01 Spotting the Signals

Meaningful deviations often emerge long before they become obvious

A late alert costs money even when the cause is simple. A retail manager sees online orders slip, support tickets rise, and SKU gaps appear in the warehouse, yet each metric still looks acceptable on its own. The problem is timing, not volume.

That is the gap anomaly detection in time series is meant to close. It adds an early layer of judgment so teams can spot when a pattern is drifting away from normal business behavior. For SMEs, that matters because small problems often surface first as weak signals, before they become visible failures.

Why the First Warning Matters

A delayed feed can look like falling demand. A payment issue can look like a conversion problem. A sensor gap can look like equipment failure. These are the kinds of edge cases that make teams doubt alerts, especially when the model flags something correctly but the source data is incomplete or stale.

The point is not to flood teams with notifications. It is to surface the right signal early enough that someone can check it while the issue is still contained.

If a metric affects revenue, service, or operations, do not wait for end-of-day reporting to notice a change.

Business leaders usually do not need more raw data. They need a way to separate normal movement from the kind of shift that deserves attention. Unlike routine monitoring, which tracks direction, anomaly detection targets unusual behavior that merits investigation.

For SMEs, the payoff is practical. Analysts can prioritize investigation, reduce wasted checking, and give teams a clearer view of what changed, when it changed, and how much confidence they should place in the alert.

02 Anomalies in Time Series

Time reveals patterns that a single snapshot of data cannot capture

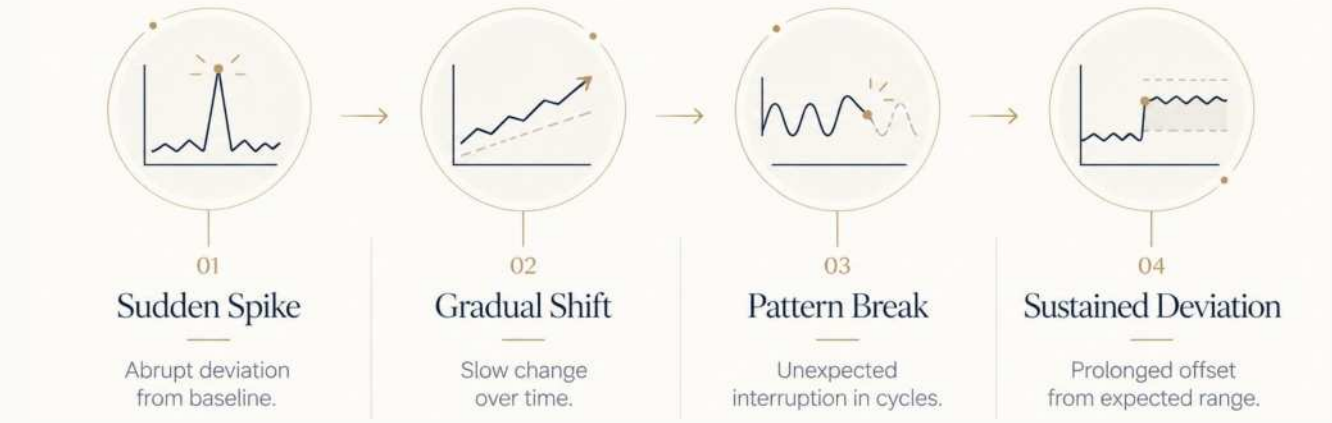
A time series is just data measured over time, like orders per hour, API latency, or daily cash collections. An anomaly is anything that breaks the pattern in a way that matters for the business, but that break doesn't always look dramatic. It might be a single sharp spike, a slow drift, a sudden pattern break, or a prolonged offset from normal behavior.

Four Patterns That Often Confuse Teams

The easiest mistake is thinking anomalies are always extreme points. In practice, they often show up as:

ANOMALY TYPE	DESCRIPTION
Sudden spikes	*Abrupt departures from the baseline, often caused by events, errors, or one-off transactions.*
Gradual shifts	*Changes that develop over time and are easy to miss when looking only at daily totals.*
Pattern breaks	*Disruptions in expected weekly or hourly cycles.*
Sustained deviations	*Values that remain outside the usual range long enough to suggest a genuine operational change.*

FIG. 3 ANOMALY TYPES IN TIME SERIES



03 Statistical, ML, and AI Detection Methods

Different approaches offer distinct ways to separate expected behavior from the unexpected

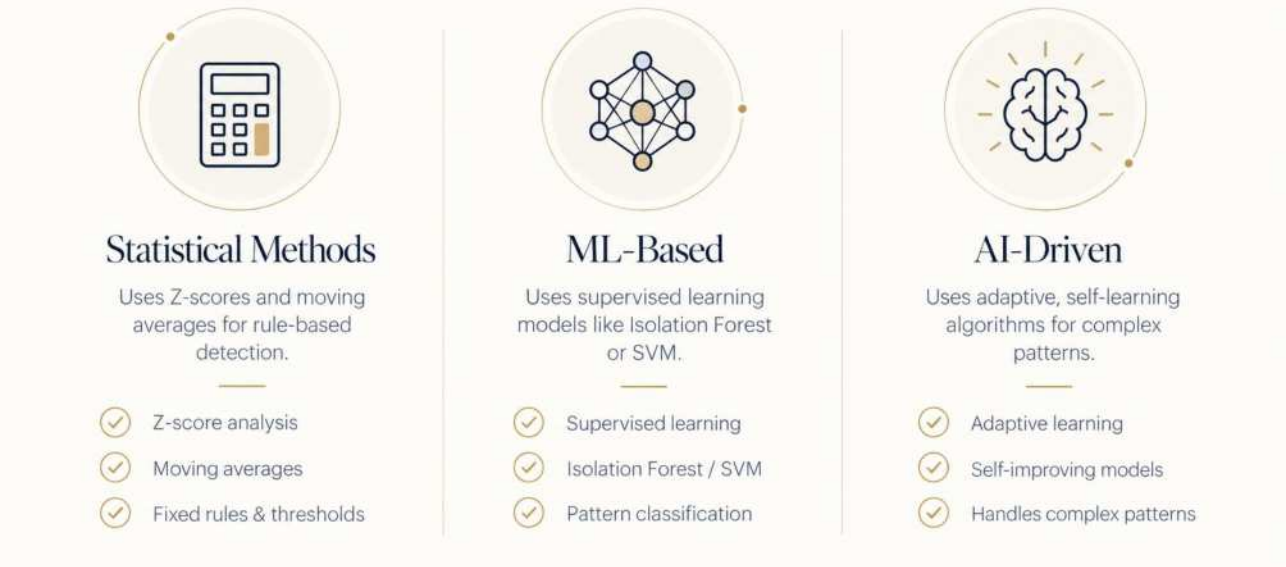
Choosing a method is less about fashion and more about fit. A simple statistical rule can be the right answer if your pattern is stable and your team needs something understandable. A more advanced model can help when the signal is messy, multivariate, or shaped by interactions that simple thresholds miss.

Three Methods, Three Purposes

Statistical methods are often the easiest starting point. They rely on rules such as moving averages, ranges, or control charts, so business teams can understand why a value was flagged. That transparency is useful when you need quick adoption and low operational overhead.

Modern AI approaches can go further by learning richer patterns directly from the data. They're useful when the structure is hard to capture with rules alone, but they also raise the bar for governance, testing, and explanation.

FIG. 4 DETECTION METHODS



How to Choose Without Overengineering

Use this practical filter:

FACTOR	WHAT TO EVALUATE	SME-FRIENDLY INDICATOR
Interpretability	*Can operations explain why it fired?*	*Clear enough for non-technical reviewers*
Setup effort	*How much data prep and tuning is needed?*	*Fast to pilot with existing data*
Pattern complexity	*Is the series simple or heavily contextual?*	*Better than a fixed threshold, but not fragile*
Maintenance	*Who updates the logic as behavior changes?*	*Fits the team that will actually own it*

A simple rule often beats a refined one when the business process is stable. An advanced model is worth it when the cost of missed anomalies is high, the pattern shifts often, or the signal depends on many variables at once.

04 Detection Performance with the Right Metrics

The right metrics reveal whether a model detects what truly matters

A model that looks accurate can still be useless in practice. That happens when the metric rewards point-by-point matches, while the core issue is an event that unfolds across a window of time. In anomaly detection, a missed portion of the anomaly interval can matter more than a slightly imperfect timestamp.

Why Point Metrics Can Mislead

Anomalies often span ranges, not single timestamps. If you only score exact point hits, you can undervalue a model that correctly catches the event but not the exact moment inside the interval. The SAS overview of time-series anomaly detection notes that range-aware measures are often a better fit, and the TSB-AD benchmark identifies VUS-PR as the most reliable measure for this setting because it reflects overlap across anomaly intervals rather than only single timestamps.

The problem goes deeper than one metric. A 2026 formal analysis examined 37 commonly used evaluation metrics and found that most satisfy only a few desirable properties, while none satisfy all of them, which helps explain why results often disagree across papers and benchmarks. The practical lesson is simple, don't trust a single score unless you know exactly what it measures.

If your alert is meant to support operations, score it the way operations experiences it, as an event, not as isolated points.

The benchmark side matters too. The TSB-AD benchmark reports 1,070 high-quality time series from 40 datasets, making it twice the size of the largest prior curated collection and four times larger than existing curated datasets, while also evaluating 40 detection algorithms spanning statistical methods and foundation models.[5] Those numbers matter because model rankings can shift under a unified setup and proper hyperparameter tuning.

5 The Elephant in the Room: Towards A Reliable Time-Series Anomaly Detection Benchmark, Qinghua Liu, December 2024

FIG. 5 KEY METRICS FOR ANOMALY DETECTION PERFORMACE



05 Batch and Streaming Anomaly Monitoring

Historical analysis and real-time monitoring serve different business needs

Implementation shapes trust. If your monitoring runs in batch, you get a cleaner retrospective view, which is fine for slow-moving processes and weekly review cycles. If your business depends on immediate response, streaming or online inference makes more sense because the alert arrives while someone can still do something about it.

Batch vs. Real-Time Monitoring

Batch pipelines are good for trend review, reporting, and historical comparison. They let you process larger windows, revisit prior periods, and reconcile results after the fact. Streaming systems are different, they focus on incoming events and fast feedback, which is why they're better for operational monitoring.

The hard part is data quality. Missing values, irregular sampling, and delayed event delivery can all create false alarms if you treat them like true business changes. Microsoft's documentation for anomaly detection in stream processing notes that gaps in a time series may mean the model didn't receive events, and it uses imputation logic to handle that case. That distinction matters for monitoring because an ingestion delay can look like a real anomaly if you don't account for it.

Practical Implementation Choices

A lot of false alarms come from the pipeline, not the process you're trying to monitor.

A stable setup usually starts with these steps:

STEP	DESCRIPTION
Clean the input stream	*Remove obvious duplicates, blank entries, and timestamp issues to prevent unnecessary noise*
Preserve event timing	*Account for irregular intervals, which can distort the shape and interpretation of a time series*
Add business context	*Include relevant events such as product releases, promotions, and maintenance periods*
Separate missing data from abnormal behavior	*Distinguish ingestion failures from genuine anomalies to avoid false alerts*

That's why feature engineering still matters. Even in automated systems, a few well-chosen derived signals can make detection steadier and easier to review. The goal is not to force every issue into a real-time alert, it's to build a monitoring path that matches how quickly the business can react.

06 Business Scenarios

Anomalies reveal different insights across business functions

A finance team reviewing AML alerts may see three small deposits just under the reporting threshold over 48 hours. That pattern can point to structuring, and it gives investigators a clearer starting point than a single large transfer would.

Retail teams face a different version of the same problem. A campaign that should lift traffic but stays flat is a signal worth investigating, especially if inventory, pricing, or site changes happened at the same time. Operations teams watch equipment, infrastructure, and data flows for the same reason. A slow decline in performance can matter more than a single spike because it often appears before service breaks down.

Anomalies Across Industries

In finance, the useful question is whether the pattern matches normal customer behavior and policy thresholds. A repeated series, a gradual drift, or a missing record can all matter if they

change the risk picture. The alert has to give compliance or risk teams enough context to decide whether review is needed.

Retail teams need different context. Inventory mismatches can point to counting errors or shrinkage, while a weak promotion can reveal a campaign, pricing, or demand issue. Operations teams use the same logic for infrastructure health, where early signs of degradation can help engineers act before users feel the impact.

Understanding Practical Use Cases

Start with the business decision, then map the detection problem:

KEY QUESTION	DESCRIPTION
What needs early warning?	*Identify critical areas to monitor, such as revenue, compliance, service performance, or uptime*
What counts as a real event?	*Define meaningful anomalies, such as spikes, data gaps, sustained shifts, or process breaks*
Who acts on the alert?	*Assign responsibility to the relevant teams, such as finance, store operations, support, or engineering*
How fast must the response be?	*Establish the required response time, from same-day review to immediate intervention*

That framing keeps anomaly detection tied to action. A model can score well and still miss the mark if the alert arrives without enough context for the team that has to respond. Business trust grows when the alert matches a real workflow and the edge cases, like short fraud patterns, flat promotions, or slow equipment drift, are easy to explain.

07 Choosing the Right Tools

The best technology depends on the data, the problem, and the speed of response required

A team can have a strong anomaly model and still struggle in production if the surrounding tooling is hard to keep running. Analysts often need flexibility for custom checks, while engineers need control over data pipelines and alert logic. Open-source libraries can suit that setup. Platform tools work better when

the goal is to reduce manual steps between raw data, detection, and review.

Key Comparison Criteria

A useful shortlist should cover these factors:

FACTOR	WHAT TO EVALUATE	SME-FRIENDLY INDICATOR
Automation	*Does it preprocess, detect, and report with limited manual work?*	*Requires minimal manual intervention after initial configuration*
Integration	*Can it connect to your current systems cleanly?*	*Fits current data flows*
Monitoring depth	*Does it support ongoing anomaly tracking, not just one-off analysis?*	*Useful beyond the pilot*
Reporting	*Can non-technical users understand the output?*	*Clear summaries, not just scores*

08 Practical Best Practices

Reliable detection starts with clear objectives, relevant data, and continuous refinement

Strong anomaly detection starts with a clear business question. If you don't define what counts as a meaningful deviation, even a good model will produce alerts that nobody trusts. The safest path is to begin with one process, one signal, and one owner who can validate whether the system is catching real events.

A Disciplined Rollout

Use these steps:

1. Pick one operational metric that has an obvious owner and a clear action path.
2. Check the data quality first, especially gaps, delays, and timestamp consistency.
3. Validate alerts against known events so you can see what the system catches and what it misses.
4. Review false alarms with the team and decide what context should suppress them.

5. Expand only after the first use case earns trust.

The mistake many teams make is optimizing for a score that looks good but doesn't reduce work or risk. A better goal is a monitoring process that helps people react sooner and with more confidence. That means making metrics, alerts, and business ownership visible together.

For SMEs, the smartest path is usually measured, not flashy. Start simple, prove that the alert map matches reality, then scale the parts that your team can support consistently.

Teaching AI to Spot the Unexpected

AI-powered anomaly detection for business data

The real promise of AI is not simply finding what looks unusual, but knowing which signals matter. Autonomous monitoring brings that distinction into everyday business operations.

BY FABIO LAURIA — CEO & FOUNDER, ELECTE

A finance manager notices an invoice that doesn't resemble anything the company usually buys. A systems administrator sees traffic behaving strangely overnight. A retail manager spots refunds that look harmless individually but suspicious when viewed together. In each case, a person is using judgment to identify an unexpected signal in familiar data.

AI-powered anomaly detection turns that instinct into a repeatable monitoring process. It examines transactions, operational metrics, security events, and other business data, then highlights patterns that differ meaningfully from the expected baseline. The global anomaly detection market is projected to reach USD 7.63 billion in 2026 and USD 16.63 billion by 2031, representing a projected 16.86% CAGR, with Asia Pacific identified as the fastest-growing region.

01 Why Anomaly Detection Matters

The earliest signs of trouble are often hidden in ordinary data

A finance team can review an unusual supplier invoice, question a sudden sales decline, or investigate a service that slows outside normal hours. That approach works while the volume is manageable. As transactions, events, metrics, and user actions multiply, people cannot inspect every signal consistently.

With AI-powered anomaly detection, manual review becomes a more continuous process. The system learns the patterns your team considers normal, flags unusual observations, and brings the most relevant signals to their attention. Rather than deciding whether an event is harmful, it helps staff determine where human judgment is needed first.

> *An alert is useful only when someone can understand it, verify it, and take an appropriate action.*

The technology now supports continuous monitoring across finance, retail, security, and IT operations, rather than serving only as an isolated statistical experiment. Its value comes from connecting detection to the work that follows. A score without business context is like a smoke alarm without a way to check which room is affected.

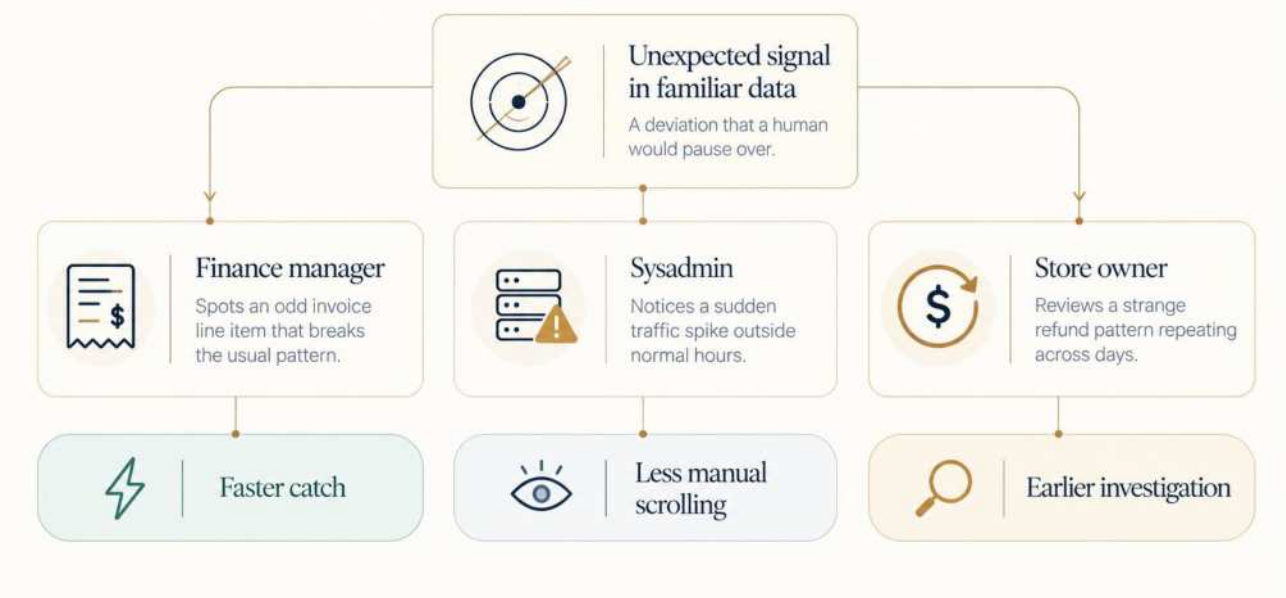
The Value of Early Detection

A detector may surface a sales pattern before it appears in a monthly report. It may group unusual access events that deserve review, or distinguish a normal peak from a deviation by considering the hour, day, customer segment, or location.

For an SME, the practical benefit is less manual scrolling, faster investigation, and more consistent decision-making. The strongest implementations connect four disciplines:

- **Algorithm selection:** Choose a method that fits the shape and stability of your data.
- **Metric selection:** Measure performance according to the cost of missed events and false alarms.
- **Deployment design:** Send scores into the systems where staff review and act on alerts.
- **Ongoing tuning:** Adjust thresholds as customer behaviour, products, seasons, and processes change.

FIG. 6 WHY ANOMALY DETECTION MATTERS



The central idea is simple: anomaly detection is a connective layer between raw operational data and trustworthy decisions. The model identifies a deviation, while your data definitions, workflow, and staff verification determine whether that signal leads to a useful action. For smaller teams, integration and context often matter more than choosing the most advanced algorithm.

02 What Counts as an Anomaly

Not every unusual event is meaningful, and context makes the difference

An anomaly is a data point, pattern, or sequence that differs meaningfully from what you expect in a specific situation. “Meaningfully” matters. A large order may be normal for one customer segment and suspicious for another. A high server load may be expected during a planned campaign but unusual during quiet hours.

Consider several examples:

- A single $12,000 refund stands apart from an average order value of $40.
- A server CPU reading holds near 95% during off-hours, even though the service normally runs at that time.
- A login from an unrecognised geography occurs at 3 a.m.
- A shipping-address change is followed by a high-value purchase.

The values in these examples are illustrative business scenarios, not universal thresholds. Your detector needs a baseline created from your own processes, customers, systems, and operating calendar.

Identifying Anomaly Patterns

Practitioners usually classify anomalies before choosing a model. The classification helps you avoid applying a point-based detector to a sequence problem or using a global threshold where context determines meaning.

Point anomalies involve one observation that stands apart from nearby or historical values. A sudden transaction spike, an isolated refund, or an unexpected sensor reading can fit this category. The detector focuses on the individual observation and its distance from the baseline.

Contextual anomalies are normal in one setting but unusual in another. Beachwear sales may be expected during warm-weather demand and unusual in December, depending on the business and market. A server load that is routine during a scheduled batch job may be concerning at night. Contextual detection requires features such as time, location, customer type, campaign status, or operational state.

Collective anomalies emerge from a group of observations. Each event may look ordinary, but the sequence creates concern. Slow credential probing across many endpoints, repeated low-value deposits, or several refunds associated with a changing account profile can form a collective anomaly.

This distinction changes the technical design. Point anomalies may work with single-row features. Contextual anomalies require the model to understand conditions around the observation. Collective anomalies need sequence, window, relationship, or graph features.

FIG. 7 WHAT COUNTS AS AN ANOMALY IN BUSINESS DATA



Before selecting a technique, write down what normal means, which context changes that meaning, and what sequence would make an event suspicious. This short exercise often improves a project more than switching between models.

03 How Anomaly Detection Algorithms Work

Behind every detected anomaly lies a method for recognizing what falls outside expectations

Anomaly detection algorithms answer a common question in different ways: how far does new behaviour depart from the expected pattern? The right choice depends on data cleanliness, time structure, dimensionality, and how much explanation investigators need.

Three Methods, Different Strengths

Statistical methods establish a mathematical baseline. A z-score can identify an observation that sits far from the historical mean,

Grubbs' test can assess an extreme value under suitable assumptions, and EWMA control charts can track changing averages over time. These approaches are quick and interpretable, but they work best when the data is relatively clean, the distribution is reasonably stable, and the operating pattern doesn't shift dramatically.

Machine learning methods learn a representation of normal behaviour from historical data. Isolation Forest isolates unusual observations through random partitions, One-Class SVM learns a boundary around expected examples, and autoencoders flag observations that they reconstruct poorly. These methods are useful when you have many interacting features and few reliable fraud or failure labels.

Time-series techniques model trend and seasonality explicitly. ARIMA can model relationships among past values and residuals, Prophet can represent recurring calendar patterns, and LSTM forecasters can learn complex sequences when you have sufficient data and the operational capacity to support a more involved model.

ALGORITHM FAMILY	REPRESENTATIVE TECHNIQUE	DATA REQUIREMENTS	BEST-FIT BUSINESS PROBLEM
Statistical	*z-score, Grubbs' test, EWMA*	*Clean, relatively stable numerical data*	*Sensor monitoring or simple KPI tracking*
Machine learning	*Isolation Forest, One-Class SVM, autoencoder*	*Fast to pilot with existing historical data and limited labels*	*Transaction monitoring or user behaviour analytics*
Time series	*ARIMA, Prophet, LSTM forecaster*	*Ordered observations with trend or seasonality*	*Revenue, traffic, or infrastructure metrics*

The same dataset can support more than one approach, but the operational trade-offs differ. Statistical methods are easier to explain. Machine learning can capture relationships that simple rules miss. Time-series models are stronger when the calendar shapes expected behaviour.

Industrial evaluation has become more demanding for similar reasons. The original MVTec AD benchmark contains more than 5,000 high-resolution images across 15 object and texture categories, while MVTec AD 2 adds eight new anomaly-detection scenarios and more than 8,000 high-resolution images.[6] These benchmarks show why image-level scores alone aren't enough for

6 Industrial anomaly detection benchmark dataset, MVTec

production inspection. Teams also need to test domain shift, multiple views, production variation, and fine-grained localization.

04 Choosing the Right Evaluation Metric

A model's value depends on more than how often it gets things right

Accuracy sounds reassuring, but anomaly detection usually involves an imbalanced dataset. Most observations may be normal, while the events you care about are rare. A model can therefore appear accurate while missing the very cases your team needs to find.

Suppose 99% of transactions are legitimate. A model that predicts every transaction as legitimate would achieve 99% accuracy, yet it would detect no fraud at all. This is why evaluation must connect to business cost rather than rely on one headline score.

METRIC	WHAT IT MEASURES	BEST FOR	RISK IF MISUSED
Precision	*How many flagged events are genuinely relevant*	*Website monitoring or queues where false alarms are expensive*	*Misses can remain hidden if the threshold is too conservative*
Recall	*How many relevant events the system catches*	*Fraud, safety, or security investigations where silent misses carry high cost*	*Alert volume can overwhelm reviewers*
F1 score	*A balance between precision and recall*	*Comparing models when both error types matter*	*Can hide which error is more damaging to your business*
AUROC	*How well the model separates classes across thresholds*	*General model comparison during development*	*May look strong even when the chosen operating threshold performs poorly*

A fraud team investigating high-value chargebacks may prioritise recall. Missing a real case can be more damaging than sending extra alerts for review. A website uptime team may prioritise precision because repeated false alarms interrupt engineers and reduce confidence in monitoring.

Thresholds Create Operational Consequences

Every threshold changes the workload. Lowering it may catch more unusual events, but it can also expand the investigation queue. Raising it may reduce noise while allowing subtle issues to pass unnoticed. Customer trust can also be affected if an

automated system blocks legitimate activity.

Use a precision-recall curve to examine that trade-off at different thresholds. Then choose the operating point with the people who will review alerts, because they understand queue capacity, customer impact, escalation rules, and the cost of delay.

The ADBench study evaluated 30 algorithms on 57 benchmark datasets, while the industrial-focused IM-IAD benchmark compared 19 algorithms across seven major datasets under a uniform setting.[7] The ranking changed across datasets, which supports a practical conclusion: validate models against domain-matched data and optimise for the business metric that reflects risk.

7 ADBench: Anomaly Detection Benchmark, Songqiao Han, 17 September 2022

05 Real-World Use Cases Across Industries

From finance to manufacturing, anomalies expose risks and opportunities

A useful anomaly detection system begins with a recognisable operational problem. The model matters, but the workflow determines whether anyone can act on its output.

Card Fraud

A customer account has been inactive for a long period. Suddenly, a $4,200 purchase arrives from a new device, alongside behaviour that differs from the account's established pattern. This is a contextual anomaly because the transaction's meaning depends on the account history, device, location, timing, and purchase characteristics.

A machine learning approach such as Isolation Forest can combine those features without requiring a complete set of labelled fraud examples. The human-owned step remains essential. An analyst or risk workflow should verify the signal, apply the organisation's authentication policy, and distinguish legitimate travel or device changes from account takeover.

Anti-Money Laundering

A single deposit may look ordinary. A sequence involving multiple accounts, repeated low-value transfers, timing relationships, and shared identifiers may reveal a more concerning pattern. This is a collective anomaly, and a detector needs relationship or sequence features rather than only transaction-level values.

A clustering approach can surface groups of accounts with similar or connected behaviour. Investigators still need to review the underlying records, document the rationale, and follow applicable legal and compliance procedures. Anomaly scores support triage, but they don't prove criminal activity.

An anomaly alert is an investigative signal, not a legal conclusion. Financial services teams should validate outputs with qualified compliance professionals and follow applicable regulations.

SaaS Operations

A software platform's overall latency may remain within a familiar range while one microservice gradually drifts above its rolling baseline. A contextual time-series model can compare the service with its own historical behaviour, account for traffic conditions, and raise an alert before customers report a problem.

The operations team owns the verification step. Engineers should inspect deployment changes, dependencies, logs, traces, and infrastructure conditions before escalating or rolling back. A model can identify where behaviour changed, but it can't independently establish the root cause.

These examples also show why one universal detector is unlikely to serve every workflow. Fraud depends on user and transaction context. AML depends on relationships and sequences. Operations depends heavily on time, dependencies, and system state.

06 Common Pitfalls

Even sophisticated models can fail when assumptions, data, or context are overlooked

Many projects fail after a promising offline evaluation. A team trains a model, sees 0.95 AUROC on a clean test set, and assumes deployment is nearly complete. Production then introduces a new payment processor, holiday seasonality, duplicate customer IDs after a CRM migration, missing fields, and behaviour the training data never represented.

The failure isn't necessarily the algorithm. The pipeline lacks operational context. A detector can't interpret a post-maintenance vibration pattern if maintenance logs sit in another system. It can't separate an expected campaign surge from a genuine problem if campaign status isn't part of the feature set.

The Production Failure Pattern

- **Unclear event schemas:** Teams use different definitions for orders, refunds, users, incidents, or assets.
- **Weak labels:** Investigators may record outcomes inconsistently, so feedback can't reliably improve the model.
- **Missing feedback loops:** The system raises alerts, but nobody records whether each alert was useful.
- **Unmonitored drift:** Customer behaviour, products, suppliers, and infrastructure change over time.
- **Unexplained decisions:** Staff can't tell why a transaction or user was flagged, creating governance concerns.

Cybersecurity adds another limitation. Anomaly-based systems learn normal behaviour from historical data, so they can struggle with zero-day or polymorphic activity that lacks a stable pattern. A company should therefore combine anomaly detection with rules, threat intelligence, access controls, and human review rather than treating one model as complete protection.

AI governance also applies when the detector monitors AI systems. Recent coverage reports that European organisations trail the global benchmark in AI anomaly detection capability, with France at 32%, Germany at 35%, and the UK at 37%, compared with 40%

globally These figures point to an emerging control problem: companies increasingly need to monitor AI usage, model behaviour, abnormal access, and policy violations, not only traditional business data.

Human-in-the-loop review isn't a temporary weakness. It's a permanent design requirement for systems that influence customers, payments, safety, compliance, or access.

FIG. 8 WHY MOST ANOMALY DETECTION PROJECTS QUIETLY FAIL



07 Tuning Best Practices

Reliable detection requires balancing sensitivity, precision, and business relevance

SMEs usually weigh three deployment paths. A hosted SaaS platform can shorten setup and reduce infrastructure work, but it may limit control over models, data handling, and configuration. An in-house build using open-source libraries such as PyOD or scikit-learn offers more control, but requires engineering, monitoring, security, and maintenance capacity.

A hybrid approach separates responsibilities. A managed service can handle scoring and infrastructure while the business owns alert routing, investigation rules, and review records. That model often suits teams that want to test the value quickly without giving up control over operational decisions.

DEPLOYMENT PATH	STRENGTH	TRADE-OFF	SUITABLE STARTING POINT
Hosted SaaS	*Faster setup and less infrastructure work*	*Less control over implementation and data flow*	*Teams validating an initial use case*
In-house open source	*Flexible models and full technical control*	*Greater engineering and maintenance burden*	*Teams with strong data and engineering capacity*
Hybrid	*Managed scoring with business-owned review workflows*	*Requires clear ownership across the boundary*	*SMEs balancing speed with governance*

A Practical Deployment Playbook

i. Start with one high-signal stream

Choose a workflow where missed anomalies already create visible pain, such as refunds, stock movement, payment events, or service latency. Avoid combining every available source in the first release.

ii. Establish a baseline before alerts

Observe normal behaviour and document business conditions that change it. A baseline should include relevant context such as time, customer segment, campaign status, maintenance activity, or service version.

iii. Use adaptive bands where appropriate

Percentile bands can reflect the observed range better than a fixed cutoff, especially when a metric varies by time or operating condition. Don't assume a percentile threshold is automatically correct. Validate it against real investigations.

iv. Route alerts to a shared queue

Include the anomaly score, affected entity, relevant features, comparison baseline, timestamp, and any known contextual event. Reviewers should be able to understand why the system raised the alert without opening several disconnected systems.

v. Capture analyst feedback

Record whether an alert was useful, expected, duplicate, or caused by a data issue. That feedback becomes evidence for threshold changes and future model selection.

vi. *Review false positives weekly*

Alert fatigue is one of the fastest ways to lose trust in a good detector. Remove noisy fields, adjust thresholds, group related alerts, or change the model when the queue becomes unmanageable.

vii. *Document assumptions and retraining decisions*

Keep a record of what the model considers normal, which data it uses, what events were excluded, and when behaviour changed. This supports auditability and helps new team members interpret alerts.

08 The Essentials

Effective anomaly detection starts with the right data, clear priorities, and meaningful signals

Anomaly detection works best when you treat it as an operating process rather than a model purchase. The detector identifies unusual behaviour, but your team defines normal, evaluates risk, verifies alerts, and decides what action follows.

Keep these principles in view:

- **Context comes first.** A number becomes meaningful when you compare it with the right customer, time period, process stage, location, or system state.
- **Data quality beats algorithm choice.** Consistent schemas, reliable identifiers, useful labels, and connected business context often matter more than moving from one advanced model to another.
- **Metrics should reflect consequences.** Use recall when missed events carry serious risk. Favour precision when false alarms consume scarce attention. Use F1 or AUROC as supporting evaluation tools, not substitutes for operational judgment.

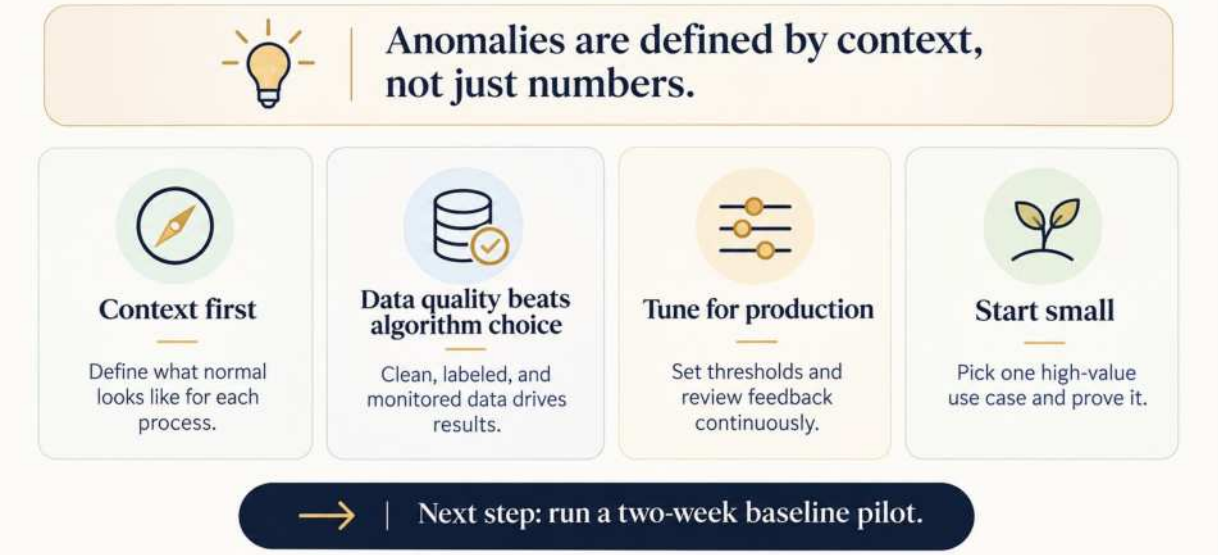
- **Tuning is continuous.** Thresholds, queues, feedback, and model assumptions need regular review as the business changes.
- **Start with one valuable workflow.** A focused pilot creates clearer evidence than a broad rollout across disconnected data sources.

A Sensible First Pilot

Choose one process where missed anomalies cause real financial, operational, security, or customer pain. Document the expected behaviour, connect the required context, observe the baseline, and ask the people who investigate exceptions to define what a useful alert looks like.

Then measure more than model performance. Track whether reviewers understand alerts, whether they can act quickly, whether false positives crowd out important cases, and whether the system exposes gaps in your data pipeline.

FIG. 9 NEXT STEPS FOR YOUR TEAM



THE ELECTE QUARTERLY

ELECTE.

The house journal of ELECTE, published four times a year from Milan

PUBLISHED BY	ELECTE S.R.L. —Via Montenapoleone 8, 20121 Milan (MI)
EDITOR	Fabio Lauria
THIS ISSUE	No. 2 — November 2026 — *On distinguishing meaningful signals from noise*
CORRESPONDENCE	hello@electe.net
COMPOSITION	Set in Playfair Display, New Times Romans and Geist
REGISTRATION	VAT IT 12771670960 — REA: MI-2682918
ISSN	3103-8743