International Journal of Artificial Intelligence, Data Science, and Machine Learning
Grace Horizon Publication | Volume 5, Issue 4, 245-265, 2024
ISSN: 3050-9262 | https://doi.org/10.63282/3050-9262.IJAIDSML-V5I4P123

Original Article

Fraud–AML Convergence: Integrating Fraud and AML Detection, Shared Typologies, and Unified Case Management.

Mallikarjun Reddy Gouni
University of Illinois Springfield.

Abstract *- Fraud and anti–money laundering (AML) programs have historically evolved as separate control functions, often operating on different data, tools, and investigative workflows. However, converging threat landscapes where scams, synthetic identities, mule networks, account takeovers, and laundering chains intersect are forcing financial institutions to rethink fragmented detection and response models. This paper examines the concept of fraud–AML convergence as an integrated approach to identifying illicit activity across the full customer and transaction lifecycle. It analyzes how shared typologies (e.g., authorized push payment fraud feeding mule accounts, trade-based laundering enabled by invoice manipulation, and crypto-fiat layering via fraudulent onboarding) can be modeled through common risk signals, entity resolution, and network analytics. The study further evaluates architectural and operational options for integrating fraud and AML detection, including unified data layers, feature stores, shared rules-and-ML pipelines, and cross-domain alert prioritization. A central focus is unified case management: consolidating alerts, evidence, and investigator actions into a single workflow that reduces duplication, improves triage accuracy, and strengthens auditability and regulatory reporting. The paper also discusses key implementation challenges governance, explainability, privacy constraints, and model risk management alongside practical success metrics such as reduced false positives, faster time-to-disposition, and improved suspicious activity conversion. By presenting an integrated typology-to-workflow framework, this research offers a roadmap for institutions seeking more resilient, efficient, and intelligence-driven financial crime operations.*

Keynotes *- Fraud–AML Convergence, Anti-Money Laundering (AML), Fraud Detection, Financial Crime Compliance, Financial Crime Risk Management, Integrated Risk Framework, Shared Fraud & AML Typologies, Suspicious Activity Monitoring, Transaction Monitoring Systems,*

1. Introduction

Financial institutions face an increasingly blended financial crime environment in which fraud and money laundering are no longer cleanly separable. Traditional program structures fraud teams focused on preventing customer harm and near-real-time losses, and AML teams focused on detecting laundering and meeting regulatory obligations were built for an era when threat patterns, data sources, and investigative workflows were more distinct. Today, those boundaries are routinely crossed. Scams and account takeover events can rapidly evolve into laundering activity through mule accounts; synthetic identity fraud may enable large-scale placement and layering; and digital payments, instant settlement, and crypto on/off-ramps have accelerated how quickly criminal proceeds can move across channels and jurisdictions.

This convergence is not merely conceptual; it is operational. Many institutions still run parallel detection stacks, separate alert queues, and duplicate investigations against the same customers, accounts, or transaction chains. Fragmentation creates blind spots (e.g., fraud signals not informing AML risk assessments), inefficiencies (duplicative evidence gathering and outreach), and inconsistent outcomes (a case closed as “fraud loss” in one team may still reflect suspicious activity under AML expectations). As criminals adopt networked operating model’s mule recruitment, coordinated identity abuse, and cross-platform laundering defending with siloed controls becomes increasingly misaligned with the nature of the threat.

Fraud–AML convergence refers to the strategic and technical integration of fraud prevention and AML detection into a unified financial crime capability. This approach seeks to harmonize data and analytics across domains, align typologies and risk indicators, and streamline investigations through unified case management. Practically, convergence aims to connect the full lifecycle of illicit activity: from onboarding and identity compromise, to transaction manipulation, to placement, layering, and integration of criminal proceeds. A converged model emphasizes shared entity-centric intelligence (customers, devices, accounts, counterparties, beneficiaries), network analytics that detect organized behavior, and cross-domain alerting that prioritizes the most harmful and suspicious activity regardless of whether it is initially labeled “fraud” or “AML.”

At the heart of convergence is the recognition that fraud and AML share a growing set of typologies and underlying enablers. Authorized push payment scams, for example, commonly feed mule networks that distribute funds through rapid transfers and cash-outs. Synthetic identity schemes can be used both to extract credit and to create laundering infrastructure. Trade-based manipulation, invoice fraud, and procurement collusion can obscure the source and purpose of funds while generating fraudulent gains. These overlapping patterns require integrated detection strategies that combine behavioral signals, typology-driven rules, machine learning, and graph-based approaches supported by robust identity resolution and consistent risk governance.

This paper explores the drivers, design principles, and practical implementation of fraud–AML convergence. It first maps shared typologies and illustrates how criminal behaviors manifest across products, channels, and time horizons. It then evaluates integration strategies for detection, including unified data foundations, shared feature engineering, entity resolution, and hybrid rules-plus-ML pipelines that serve both fraud and AML objectives. Finally, it examines unified case management as an operating model that consolidates alerts, evidence, investigator actions, and reporting pathways to reduce duplication, improve decision consistency, and strengthen auditability.

By presenting an integrated typology-to-workflow perspective, this research aims to provide a roadmap for institutions seeking to modernize financial crime operations improving detection effectiveness, reducing false positives, accelerating investigations, and aligning operational outcomes with both customer protection and regulatory expectations.

1.1. Background and Motivation

1.1.1. Rising Complexity of Digital Payments and Identity Abuse

The rapid digitization of financial services has expanded both the speed and surface area of financial crime. Instant payments, mobile wallets, open banking integrations, card-not-present commerce, embedded finance, and crypto on/off-ramps have reduced friction for legitimate customers but also for criminal networks. Funds can now move across accounts and jurisdictions in seconds, leaving narrow windows for detection and intervention. At the same time, customer journeys have become increasingly remote and automated (eKYC, digital onboarding, self-service account changes), which amplifies exposure to identity-based threats.

Identity abuse has evolved from simple credential theft into an industrialized ecosystem that includes synthetic identities, account takeovers, deep fake-enabled social engineering, SIM swapping, device emulation, and mule recruitment at scale. Criminals exploit weak identity proofing, reused credentials, compromised devices, and gaps between digital and physical identity controls. These tactics often serve dual purposes: they facilitate fraud losses (e.g., unauthorized transfers, chargebacks) while also creating or commandeering laundering infrastructure (e.g., mule accounts used for placement and layering). As a result, the same "identity event" can be the entry point to both fraud and AML risk—making separated detection programs increasingly misaligned with how attacks actually unfold.

1.1.2. Fragmentation: Fraud Teams vs. AML Teams

Despite converging threats, many institutions still operate fraud and AML functions as parallel systems with different objectives, timelines, and tooling:

- Fraud teams are typically optimized for *real-time prevention* and customer protection. Their detection models prioritize immediacy blocking transactions, stepping up authentication, and minimizing direct losses and customer harm. Signals such as device fingerprinting, behavioral biometrics, authentication telemetry, and velocity patterns are central, and success is measured in prevented loss, reduced friction, and fast decisioning.
- AML teams are generally structured around *post-event monitoring* and regulatory compliance. Their systems often run in batch or near-batch modes, focused on identifying suspicious patterns over longer horizons (days to months), producing alerts that support investigation and regulatory reporting. AML detection relies heavily on transaction monitoring, customer risk rating, sanctions screening, beneficial ownership context, and narrative-based casework. Success is often measured in investigation quality, auditability, and suspicious activity reporting outcomes.

This separation introduces three major problems:

- Blind spots across the crime lifecycle. Fraud controls may stop or miss events that later become laundering activity, while AML monitoring may detect "suspicious movement" without visibility into upstream fraud triggers (e.g., scam typology, social engineering indicators, or compromised identity signals).
- Duplicate alerts and inconsistent decisions. The same customer or transaction chain can generate multiple alerts in different queues, leading to repetitive evidence collection, repeated customer outreach, and conflicting conclusions (e.g., "fraud loss" vs. "suspected laundering").
- Inefficient investigations and slower response. Fragmented case management prevents investigators from seeing the full network picture devices, accounts, beneficiaries, counterparties, and linked identities reducing both triage accuracy and the ability to disrupt mule networks quickly.

Motivated by these realities, fraud–AML convergence has emerged as a strategic response: integrate typologies, data, analytics, and investigations so institutions can detect illicit activity end-to-end faster, more consistently, and with better alignment to both customer protection and regulatory outcomes.

1.2. Research Problem and Gaps

Although fraud and AML risks increasingly originate from the same actors, channels, and enabling behaviors, most institutions still address them through separate operating models. The core research problem this paper investigates is: how can financial institutions design an integrated fraud–AML capability that supports shared typologies, consistent risk assessment, and unified investigations without weakening domain-specific controls or regulatory defensibility? Two persistent gaps drive this problem: the absence of shared typologies and the lack of a consistent customer/entity risk view across domains.

1.2.1. Gap 1: Lack of Shared Typologies and Unified Investigations

A major weakness in current financial crime frameworks is that fraud and AML typologies are often documented, detected, and investigated independently. Fraud typologies typically emphasize attack vectors and loss events (e.g., social engineering, account takeover, authorized push payment scams), while AML typologies emphasize movement patterns and intent (e.g., structuring, rapid movement, layering, trade-based laundering). In practice, however, these patterns are increasingly sequential and connected. For example, scam proceeds may move through mule accounts (fraud-to-AML transition), or synthetic identities may be used to both extract value and create laundering capacity (fraud-and-AML co-occurrence).

Because typology libraries and investigative playbooks are not harmonized:

- Alerts are generated without end-to-end context. Fraud systems may flag a transaction as anomalous but not connect it to downstream laundering networks; AML systems may flag "rapid movement" but miss upstream indicators of compromise or scam inducement.
- Casework becomes fragmented. Separate investigations may be opened for the same customer, beneficiary, or device footprint. Evidence is collected repeatedly, narratives diverge, and escalation thresholds differ.
- Network disruption suffers. Organized groups exploit institutional silos: even if one domain sees part of the chain, neither team has full visibility into the actor network (mules, recruiters, compromised identities, shared devices, and repeating counterparties).

This gap creates a measurable operational impact: increased false positives, slower time-to-disposition, inconsistent outcomes, and missed opportunities to identify and disrupt criminal networks early.

1.2.2. Gap 2: Inconsistent Customer/Entity Risk Views across Domains

A second gap lies in the lack of a unified, entity-centric risk perspective. Fraud and AML systems often maintain separate "truths" about the same customer, account, device, or counterparty. These risk views diverge for several reasons:

- Different data foundations. Fraud platforms may rely on device telemetry, authentication signals, session behavior, chargeback outcomes, and real-time payment metadata. AML platforms may rely on KYC attributes, customer risk rating, transaction monitoring aggregates, beneficial ownership, and sanctions/PEP screening.
- Different identifiers and resolution logic. Fraud analytics often track devices, emails, phone numbers, and behavioral patterns; AML analytics are more account- and customer-ID centric. Without strong entity resolution, the same real-world actor can appear as multiple unrelated entities.
- Different risk scoring time horizons. Fraud models prioritize immediacy and event-level risk; AML risk scoring often emphasizes longitudinal behavior and customer lifecycle risk.

The consequences of inconsistent risk views include:

- Misaligned prioritization. A customer labeled "low risk" in AML could be actively involved in fraud-related mule activity, while a fraud "high risk" device network may never elevate the AML risk rating of linked accounts.
- Inconsistent treatment and controls. One team may apply enhanced due diligence or restrictions while the other continues business-as-usual, creating control gaps and uneven customer outcomes.
- Weak feedback loops. Confirmed fraud outcomes (e.g., scam victim confirmation, device compromise) may not flow into AML models; confirmed AML outcomes (e.g., suspicious activity disposition) may not strengthen fraud detection features or blocklists.

1.3. Resulting Research Gap and Need for an Integrated Framework

These gaps indicate the need for an integrated framework that (1) maps shared fraud–AML typologies into common signals and detection logic, (2) creates a consistent entity resolution and risk scoring layer, and (3) supports unified case management to consolidate alerts, evidence, and investigative actions. Yet institutions face real constraints governance

boundaries, regulatory expectations, privacy and data minimization, explainability, and model risk management that complicate convergence.

Accordingly, this paper addresses the problem by proposing a typology-driven convergence approach that aligns detection and investigations around a single entity-centric financial crime view, enabling consistent decisions, faster disruption of criminal networks, and stronger defensibility across both fraud prevention and AML compliance obligations.

2. Conceptual Foundations: Fraud vs AML

Fraud and AML are often discussed under the umbrella of "financial crime," yet they originate from different mandates, success criteria, and operating rhythms. Understanding these distinctions and where they break down in modern threat environments is essential before proposing convergence. This section clarifies core definitions, contrasts key constraints, explains why convergence is accelerating, and synthesizes prior work to establish what exists and what remains missing.

2.1. Definitions and Goals

Fraud refers to deceptive activity intended to obtain an unlawful benefit or cause financial harm. In institutional settings, fraud programs primarily aim to prevent or mitigate losses and customer harm, typically by intervening in real time or near real time. Their mission is action-oriented: stop suspicious activity before it settles, reduce chargebacks and write-offs, protect accounts, and minimize friction while maintaining conversion.

Anti–money laundering (AML) focuses on identifying and deterring the movement and concealment of illicit proceeds and ensuring compliance with regulatory obligations. AML programs aim to detect and investigate suspicious activity and meet reporting requirements (e.g., suspicious activity reports), typically through batch or near-real-time monitoring that evaluates patterns across longer horizons. Their mission is evidentiary and governance-heavy: establish suspicion thresholds, document investigative rationale, and produce audit-ready outcomes.

While fraud is frequently framed as "loss prevention" and AML as "compliance," both ultimately attempt to identify high-risk actors and behaviors. The difference is that fraud optimizes for speed and intervention, whereas AML optimizes for defensible suspicion and reporting.

2.2. Differences in Constraints

Even when fraud and AML use similar signals, their decision environments differ. These constraints shape detection design, model choice, thresholds, and workflow.

- Latency requirements. Fraud decisions often occur at authorization time (milliseconds to seconds) and must support step-up authentication, transaction holds, or declines. AML monitoring can tolerate slower cycles but must capture broader behavioral context across time windows and channels.
- Explainability expectations. Fraud models can be highly predictive and operationally effective even when partially opaque, as long as they support rapid action and do not systematically harm legitimate customers. AML decisions, however, must often be supported by explicit, traceable rationales tied to typologies, risk factors, and regulatory standards.
- Auditability and documentation. Fraud outcomes are frequently measured operationally (loss prevented, friction, false declines), with a focus on performance and customer impact. AML requires case narratives, evidence trails, consistent dispositions, and reproducible logic to satisfy regulators, internal audit, and model risk management.
- Thresholds and tolerances for false positives. Fraud teams often accept a certain false positive rate if it reduces losses, but excessive false positives create customer friction and revenue leakage ("false declines"). AML teams must manage alert volumes to avoid "alert fatigue," yet they also face the risk of under-reporting. The cost functions differ: fraud false positives can be immediately visible to customers; AML false negatives can become regulatory findings.
- Governance and change control. Fraud rules and models may iterate quickly (daily/weekly tuning), responding to adversarial behavior. AML change control is typically slower and more formal, emphasizing validation, documentation, and policy alignment.

These differences help explain why many institutions historically separated fraud and AML. They also highlight the challenge of convergence: integration must respect distinct constraints while enabling shared intelligence and coherent decisions.

2.3. The Convergence Drivers

Despite different mandates, several forces are pushing fraud and AML toward operational and analytical convergence:

- Shared data sources. Modern detection increasingly depends on overlapping data: onboarding/KYC attributes, transaction metadata, device and session telemetry, authentication logs, customer communications, geolocation, and counterparty networks. When these are split across domains, each function sees only part of the risk picture.

- Shared actors and networks. Organized crime groups do not separate "fraud teams" and "laundering teams." Mule recruiters, synthetic identity operators, and scam rings frequently run end-to-end pipelines from victim acquisition to cash-out—across channels.
- Shared channels and speed. Instant payments, P2P transfers, and real-time settlement compress the time available to act. Laundering activity can occur immediately after fraud, meaning AML patterns emerge too late unless informed by fraud signals upstream.
- Shared typologies. Many typologies now span both domains: scam proceeds flowing through mule accounts, account takeover enabling rapid withdrawals and layering, synthetic identities enabling both fraud extraction and laundering infrastructure, and crypto-fiat movement combining deception, account abuse, and layering.

Together, these drivers motivate a shift from siloed, function-specific monitoring to a single entity-centric view of risk supported by cross-domain analytics and unified investigations.

2.4. Prior Work and Industry Context

Existing research and industry practice has made meaningful progress on components of convergence, but these advances often remain compartmentalized.

A widely recognized issue is alert fatigue: both fraud and AML systems can generate large volumes of low-quality alerts that overwhelm investigators. Industry responses include improved calibration, better segmentation, triage automation, and risk-based prioritization. Yet much of this optimization happens within each domain independently, so duplicated alerts and fragmented evidence persist when the same underlying network triggers both fraud and AML monitoring.

In parallel, AML detection has increasingly adopted graph-based methods and network analytics to reveal mule rings, layering structures, and hidden beneficial relationships. Fraud programs have long used link analysis for device-to-account relationships, merchant abuse, and coordinated attacks. The shared insight is that entity networks often carry more signal than isolated transactions. What is often missing is a consistent, cross-domain graph that unifies identity, devices, accounts, counterparties, and behaviors across fraud and AML paired with governance that allows both teams to act on it.

Another active area is identity resolution (entity resolution) and feature unification. Institutions are investing in mastering customer identity across email/phone/device/account identifiers, as well as resolving businesses, beneficial owners, and counterparties. This is foundational for convergence because fraud frequently tracks non-traditional identifiers (device/session), while AML typically anchors to customer/account identifiers. The gap remains that many implementations stop at partial linkage and do not translate resolved identities into a shared risk view and shared investigative workflows.

Similarly, real-time risk scoring has matured in fraud prevention, and "near-real-time AML" is expanding as payment rails accelerate. However, organizations often deploy separate scoring engines, thresholds, and queues—resulting in inconsistent treatment decisions for the same actor. The missing element is an integrated approach that reconciles differing latency and explainability needs: real-time interdiction for high-risk events, paired with defensible AML reasoning and longitudinal monitoring for network suspicion.

Finally, many institutions are modernizing case management workflows, introducing better evidence capture, workflow routing, and investigator tooling. Yet case management is frequently implemented as separate systems (or separate workflows within the same platform) for fraud and AML. Without a unified case layer, analysts cannot easily see upstream and downstream context, causing duplicated outreach, inconsistent dispositions, and weaker network disruption.

Overall, the literature and practice suggest that the building blocks exist network analytics, identity resolution, real-time scoring, and workflow tooling but they are rarely combined into a cohesive, typology-driven operating model. Convergence requires not only shared analytics, but also shared typologies, consistent risk semantics, and unified investigations that preserve regulatory defensibility while improving speed and effectiveness.

Table 1: Fraud vs AML Comparison Matrix

Dimension	Fraud Detection	AML Detection
Primary objective	Prevent/mitigate losses and customer harm	Detect suspicious activity; comply with reporting obligations
Typical timing	Real-time / near real-time	Batch / near real-time (increasingly faster for instant rails)
Unit of analysis	Transaction/session/event; customer/account	Customer/account and transaction patterns over time; networks
Common signals	Device/session telemetry, authentication, velocity, behavioral patterns, chargebacks	KYC/CDD, transaction monitoring aggregates, counterparty patterns, sanctions/PEP, beneficial

		ownership
Common methods	Rules + supervised ML, anomaly detection, behavioral analytics	Rules/scenarios, statistical monitoring, anomaly detection, graph analytics, segmentation
Decision tolerance	Lower tolerance for customer friction; balance false declines vs loss	Balance alert volume vs missed suspicion; regulatory defensibility prioritized
Explainability needs	Operationally actionable; may tolerate partial opacity	High: rationale, typology mapping, and reproducible evidence expected
Governance cadence	Rapid tuning and iteration	More formal change control, validation, documentation
Key workflows	Step-up auth, blocks/holds, customer contact, recovery	Investigation, disposition, escalation, reporting, audit trail
Typical KPIs	Loss prevented, fraud rate, false decline rate, approval rate, time-to-decision	Alert quality, SAR conversion/quality, time-to-disposition, audit outcomes, investigator productivity

3. Shared Typologies and Cross-Domain Patterns

Fraud–AML convergence becomes practical when an institution can describe criminal behavior consistently across teams, translate those descriptions into measurable signals, and investigate outcomes within a single narrative. Typologies are the bridge between "what criminals do" and "what systems can detect," connecting policy language, model features, and investigator decisioning.

3.1. Why Typologies Matter

Typologies are structured descriptions of recurring financial crime behaviors (e.g., "money mule network," "synthetic identity onboarding," "trade-based laundering"). They matter because they create a **common language** across stakeholders who otherwise speak different dialects:

- Model builders need typologies to define target behaviors, label strategies, and feature hypotheses.
- Rules engineers need typologies to encode scenarios and thresholds in interpretable logic.
- Investigators rely on typologies to triage alerts, gather evidence, and write coherent narratives.
- Compliance and regulators expect typologies to underpin suspicious activity rationale and demonstrate risk coverage.
- Operations leaders use typologies to define metrics: false positives by scenario, time-to-disposition, conversion rates, and control effectiveness.

Without shared typologies, fraud and AML teams end up optimizing locally building strong models for their own alert streams while missing cross-domain continuity (e.g., "this scam payout is the start of a mule ring," or "this mule ring is fed by repeated ATO events").

3.2. Typology Mapping Framework

To unify fraud and AML typologies, this paper adopts a layered mapping model that moves from high-level actors to measurable outcomes:

Actor → Capability → Channel → Technique → Indicator → Outcome:

- Actor: Who is behind the activity? (individual, organized ring, insider, mule recruiter)
- Capability: What enables them? (synthetic identity creation, credential compromise, document forgery, automation/bots, social engineering)
- Channel: Where does activity occur? (instant payments, cards, ACH/wires, merchant refunds, crypto exchange, trade finance)
- Technique: How is the behavior executed? (ATO, APP scam persuasion, structuring, rapid pass-through, invoice manipulation)
- Indicator: What can be observed and measured? (device reuse, beneficiary churn, network motifs, velocity spikes, inconsistent identity attributes)
- Outcome: What is the result? (fraud loss, mule network formation, laundering placement/layering, chargeback exploitation, SAR filing)

This structure supports convergence because fraud teams often start at technique/outcome (loss events), while AML teams often start at channel/outcome (suspicious movement). The layered model aligns both into a single narrative chain and makes feature engineering explicit.

3.3. Shared and Adjacent Typologies

Below are high-value typologies where fraud and AML either directly overlap or appear in adjacent stages of the same criminal pipeline.

3.3.1. Synthetic Identity and Identity Theft

Cross-domain pattern: fraud onboarding + laundering infrastructure creation

- Fraud angle: synthetic identities used to open accounts, obtain credit, exploit promotional offers, or enable first-party fraud behaviors.
- AML angle: the same identities can be used to create "clean-looking" accounts for mule recruitment, pass-through activity, and layering.
- Convergence signal: identity anomalies that predict both immediate fraud risk and future laundering capacity (e.g., repeated identity attributes across multiple accounts, inconsistent documentation, shared devices).

3.3.2. Money Mule Networks (APP Scams, Card Fraud Proceeds, Account Takeover)

Cross-domain pattern: victim funds or stolen value → mule account → rapid distribution/cash-out

- Fraud angle: APP scams and ATO generate transfers to new beneficiaries; card fraud proceeds can be monetized via refunds or cash-outs.
- AML angle: mule networks exhibit pass-through behavior, fan-in/fan-out flows, and rapid movement across linked accounts.
- Convergence signal: link analysis across payees/beneficiaries, shared devices, shared contact details, and repeated cash-out endpoints.

3.3.3. Smurfing/Structuring Linked to Fraud Proceeds

Cross-domain pattern: breaking stolen/scammed funds into smaller amounts to avoid thresholds and accelerate cash-out

- Fraud angle: attackers may split transfers to multiple beneficiaries or repeatedly initiate smaller transactions to bypass controls.
- AML angle: classic structuring indicators can reflect laundering intent, especially when linked to known fraud triggers upstream.
- Convergence signal: sub-threshold repetition, short time gaps, repeated rounding patterns, and networked dispersal.

3.3.4. Trade-Based Laundering Vs Invoice Fraud Overlaps

Cross-domain pattern: falsified trade documentation used to justify movement of value

- Fraud angle: invoice fraud, procurement fraud, and collusive billing distort legitimate payment purposes and divert funds.
- AML angle: over/under-invoicing, phantom shipments, and circular trade can mask illicit proceeds and enable layering.
- Convergence signal: inconsistencies across invoices, shipping data, counterparty relationships, and unusual payment terms especially when counterparties are network-linked.

3.3.5. Crypto On/Off-Ramp Laundering and Scam Proceeds

Cross-domain pattern: scams generate fiat transfers → crypto purchase → rapid movement and cash-out

- Fraud angle: victims are persuaded to send funds to accounts tied to exchanges, OTC brokers, or "investment platforms."
- AML angle: on/off-ramp activity can show classic laundering behaviors: rapid conversion, peeling chains, hopping services, and cross-border movement.
- Convergence signal: repeated transfers to exchange-related beneficiaries, device/session patterns indicating coordinated cash-out, and abrupt behavior shifts immediately following high-risk communications or onboarding anomalies.

3.3.6. Merchant Collusion, Refund Fraud, Chargeback Schemes With Laundering Patterns

Cross-domain pattern: merchants or collusive actors generate synthetic refunds/chargebacks to monetize illicit value

- Fraud angle: refund abuse, friendly fraud, chargeback manipulation, collusive merchant schemes.
- AML angle: laundering can occur through inflated refunds, circular transactions, and settlement manipulation especially across connected merchant clusters.
- Convergence signal: abnormal refund ratios, repeated refund-to-different-instrument patterns, connected merchants sharing ownership/contact/device infrastructure, and fast settlement-to-cash-out flows.
- Key takeaway: These typologies are best detected not as isolated anomalies but as linked behaviors across entities and time which argues directly for entity resolution, graph features, and unified investigations.

3.4. Typology-to-Features Translation

A major practical barrier is translating narrative typologies into measurable indicators that detection systems can score. The typology mapping model supports a repeatable translation pipeline:

- Typology narrative (investigator/regulatory language)

- Observable indicators (what would we expect to see?)
- Data fields (where does that evidence live?)
- Detection methods (rules, ML, graph, anomaly, hybrid)
- Case evidence (what must be captured for defensibility?)

Examples of measurable indicator families:

- Graph motifs: fan-in/fan-out patterns, intermediary hubs, short path lengths between entities, circular flows, shared endpoints.
- Velocity patterns: rapid movement post-credit, bursts following onboarding, sub-threshold repetition, time-of-day anomalies.
- Device and identity reuse: same device across many accounts, recycled contact details, SIM/device swaps, repeated IP blocks.
- Beneficiary churn: many new beneficiaries, first-time payees, rotating payee set, beneficiary "lifetimes" that are short.
- Behavior shifts: abrupt changes in transaction amount/channel/counterparty after credential change, SIM swap, or onboarding.

Table 2: Typology → Indicators → Data Fields → Detection Methods

Typology	Key indicators (measurable)	Core data fields	Detection methods
Synthetic identity onboarding	attribute reuse across identities; low-history high-activity; inconsistent KYC	name/DOB/address/phone/email; doc metadata; device/IP; onboarding outcomes	entity resolution; anomaly detection; supervised ML; rules for attribute collisions
Mule network (APP/ATO proceeds)	fan-in/fan-out; rapid pass-through; shared beneficiaries; short account lifetimes	transfers, beneficiaries, timestamps; device/session; account linkages	graph analytics; community detection; velocity rules; hybrid scoring
Structuring tied to fraud proceeds	repeated sub-threshold txns; bursty timing; dispersal to many payees	amount, time gaps, payee list, channel	rules + statistical tests; sequence models; network dispersal features
Trade/invoice manipulation	invoice-payment mismatch; counterparty anomalies; circular trade links	invoice IDs, amounts, terms; counterparties; shipment refs (if available)	rules; anomaly detection; graph linkage across counterparties
Crypto on/off-ramp laundering	repeated transfers to exchange-like beneficiaries; rapid conversion behavior; post-scam bursts	beneficiary identifiers; merchant category/labels; crypto gateway refs; timing	typology rules; clustering; graph paths to known endpoints
Merchant collusion/refund laundering	abnormal refund rate; refund-to-different-instrument; connected merchant clusters	merchant ID/ownership; refund ratios; settlement flows	peer-group anomaly; graph ownership links; supervised ML

4. Data Integration for Convergence

Fraud AML convergence succeeds or fails on data. Even the best models and investigators cannot connect upstream fraud triggers to downstream laundering behaviors if identity, transactions, and context signals are fragmented across systems. This section defines the core data domains to unify, how to resolve identities across noisy identifiers, how to manage quality and lineage, and how to design a unified risk view that supports both real-time interdiction and audit-ready investigations.

4.1. Data Domains to Unify

A converged detection and investigation stack typically draws from the following domains (often owned by different teams and stored in separate platforms):

- Customer & KYC/CDD: identity attributes, verification outcomes, onboarding steps, document metadata, beneficial ownership (businesses), risk rating, customer segment.
- Accounts & relationships: account hierarchy, product holdings, linked cards/wallets, authorized users, business entity associations, account status changes.
- Transactions: payments, transfers, deposits/withdrawals, card authorizations/clearing, refunds/chargebacks, fees, cash activity, virtual asset flows (where applicable).
- Devices & sessions: device fingerprints, browser/app telemetry, session patterns, MFA events, SIM change indicators, login anomalies.

- Channels: mobile app, web, branch, call center, ATM, API/open banking, merchant POS/e-commerce.
- Merchants: merchant IDs, MCC, settlement data, refund ratios, ownership/contact/processor linkages.
- Counterparties/beneficiaries: payee bank/account, beneficiary lifecycle, shared endpoints, external account linkages.
- IP, geolocation & network telemetry: IP reputation, ASN, geolocation, impossible travel, TOR/VPN usage.
- Sanctions/PEP: screening hits, resolution status, watchlist sources, match confidence.
- Adverse media (if applicable): entity mentions, risk topics, source credibility, recency.

- Design principle: unify these domains around an entity graph, not just a transaction table. Convergence depends on linking customers, accounts, devices, counterparties, merchants, and identifiers into a consistent network.

4.2. Identity Resolution and Entity Linking

Convergence requires a shared "entity layer" that answers: who is this actor, what do they control, and what are they connected to? Identity resolution (IR) bridges the gap between fraud-centric identifiers (device/session) and AML-centric identifiers (customer/account).

Core linkages to support

- Customer ↔ account: one-to-many and many-to-many relationships, authorized users, business beneficial owners, signatories.
- Customer ↔ device: device fingerprint, app instance, cookie IDs, SIM/device changes, emulator flags.
- Customer ↔ merchant/counterparty: payee relationships, recurring beneficiaries, related merchants, shared settlement endpoints.
- Account ↔ counterparty/beneficiary: bank accounts, wallet addresses, payment handles, exchange endpoints.
- Device ↔ account/beneficiary/merchant: shared devices used for multiple identities, repeated device access to payee setup.

Handling noisy identifiers and shared devices

- Noisy identifiers: emails, phones, addresses, names, and IPs are frequently reused or obfuscated. Use normalization (standardization, parsing, transliteration), fuzzy matching, and confidence scoring rather than hard equality.
- Shared devices: households, small businesses, call centers, and public networks create legitimate sharing. Treat device sharing as a risk modifier rather than an automatic block: model *degree of sharing*, *diversity of behavior*, and *coherence of identity attributes*.
- Evidence weighting: assign weights to links (strong vs weak) based on stability and spoofability (e.g., government ID verification outcomes > email; device hardware attestation > IP).
- Temporal linking: links have lifetimes—device ownership, phone numbers, and IPs change. Store time-bounded edges so investigators can see *when* a relationship existed.

Output of IR: a canonical entity graph with:

- a unique entity key (person/business/device/merchant/counterparty),
- link confidence and provenance,
- time windows,
- supporting evidence fields.

This makes downstream detection and case management consistent across fraud and AML.

4.3. Data Quality and Lineage

A converged platform is only as strong as its weakest domain. Common failure modes and controls include:

- Missingness: incomplete KYC fields, device telemetry gaps, absent counterparty attributes.
  Controls: completeness checks, fallback features, explicit missingness flags, "data coverage" metrics per channel.
- Timeliness: delayed transaction feeds, batch KYC updates, late chargeback outcomes.
  Controls: freshness SLAs per source, watermarking, late-arrival handling, and model features that account for delayed labels.
- Duplicates & entity fragmentation: multiple customer IDs for the same person, repeated transactions, duplicated merchants.
  Controls: de-duplication rules, IR reconciliation, canonical identifiers, merge/split governance.
- Schema drift & inconsistent semantics: renamed fields, changed formats, new channel codes, altered event definitions.
  Controls: schema registry, automated contract tests, versioning, backward-compatible transformations.
- Lineage and reproducibility: inability to explain why an alert fired because inputs were overwritten or untracked.
  Controls: lineage metadata (source, timestamp, transformation version), feature snapshots, and immutable audit logs.

- Practical target: every alert should be reproducible from "feature snapshot + model/rule version + entity graph state at decision time."

4.4. Privacy and Regulatory Constraints

Convergence is not a license to centralize everything without governance. Integrated data must comply with privacy, banking secrecy, and regulatory expectations.

Key constraints and design responses:

- Purpose limitation: use data only for defined fraud/financial-crime purposes.
  Response: documented lawful basis, clear use cases, and policy-based access (who can see what, and why).
- Data minimization: collect/store only what is necessary.
  Response: tiered storage (raw vs derived), selective retention, and "privacy-by-design" feature engineering.
- Retention rules: different data types have different retention requirements and legal holds.
  Response: retention schedules by domain, auto-expiry, and legal hold overrides.
- Access controls and segregation: investigators may not need raw device telemetry; modelers may not need full PII.
  Response: role-based access control (RBAC), attribute-based access (ABAC), tokenization, and audit trails.
- Cross-border data transfer: multinational institutions must handle regional constraints.
  Response: data residency strategies, regional processing, and controlled sharing of derived risk signals vs raw data.

A converged architecture should treat privacy controls as first-class components—embedded into pipelines, feature stores, and case management.

4.5. Feature Store and Unified Risk View

To operationalize convergence, institutions increasingly build a shared feature store and an **entity risk profile** that is usable across fraud and AML.

Single "entity risk profile" should include

- Entity identifiers and linked graph context: customer ↔ accounts ↔ devices ↔ counterparties ↔ merchants (with confidence + timestamps).
- Behavioral aggregates across horizons: real-time (minutes), short-term (hours/days), long-term (weeks/months).
- Cross-domain risk signals: fraud indicators (ATO likelihood, device anomaly, beneficiary novelty) plus AML indicators (structuring risk, pass-through, network centrality, sanctions proximity).
- Disposition feedback: confirmed fraud outcomes, confirmed suspicious activity outcomes, false-positive flags, investigator notes (appropriately governed).
- Explainability pack: top contributing indicators and evidence pointers suitable for case narratives.

Architecture pattern

- Real-time stream layer supports interdiction (fraud-style latency).
- Near-real-time/batch layer supports longitudinal AML monitoring.
- Both write to the same entity-centric feature and graph layers.
- Case management consumes the unified profile to consolidate alerts and evidence.

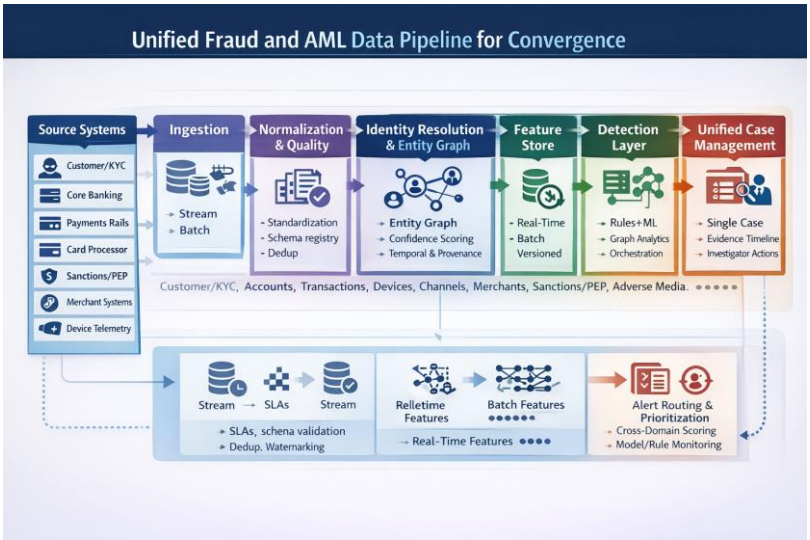

Fig 1: End-to-End Unified Fraud and AML Data Pipeline Architecture

Detection is where convergence becomes tangible: the institution decides *what to* score, what to stop, what to queue for investigation, and how to explain it. A converged architecture must support different latency requirements (fraud real-time vs AML near-real-time/batch), different evidentiary expectations, and different operational actions while still producing a unified view of risk and a single investigative narrative.

5. Detection Architecture and Analytics Integration

5.1. Architectural Patterns for Convergence

Below are four practical patterns, ordered from lowest to highest integration depth. Many institutions evolve through these patterns over time.

5.1.1. Pattern A: Shared Data + Separate Models (Fastest Governance Path)

What it is: Build a unified data layer (and entity graph) but keep fraud and AML models separate, each with its own thresholds and workflows.

- Strengths: quickest path to value; minimal disruption; easier model risk management (MRM) because each domain retains familiar controls.
- Weaknesses: still risks inconsistent risk semantics and duplicated alerts unless downstream routing is unified.
- Best for: early convergence; organizations with strict AML governance cadence and mature fraud stack.

5.1.2. Pattern B: Shared Features + Ensemble Of Fraud + AML Scores

What it is: Build a common feature store and compute multiple domain scores (fraud score, AML suspicion score, mule-network score, etc.), then combine via an ensemble or meta-model for prioritization.

- Strengths: improves consistency and prioritization without forcing a single model; supports "risk event" logic and dedup.
- Weaknesses: can create complex score interactions; requires careful calibration so the ensemble doesn't over-amplify noisy signals.
- Best for: operational convergence (shared triage/case) while preserving domain models.

5.1.3. Pattern C: Unified Multi-Task Model (Shared Encoder; Separate Heads)

What it is: One representation learning backbone (shared encoder) that learns common signals, with separate output heads for fraud loss risk, AML suspicion, mule likelihood, scam likelihood, etc.

- Strengths: captures shared structure; reduces duplicated feature engineering; can improve generalization across sparse typologies.
- Weaknesses: harder to govern and explain; label definitions differ (fraud confirmation vs AML suspicion); needs strong MRM and drift controls.
- Best for: institutions with mature data/label pipelines and strong model governance.

5.1.4. Pattern D: Graph-First Entity Risk Engine Feeding Both Fraud + AML

What it is: An entity graph is the primary detection substrate. Graph analytics produces entity/network risk scores (e.g., mule cluster score, centrality anomalies, proximity to known bad), which feed both fraud interdiction and AML monitoring.

- Strengths: naturally aligns with shared typologies; strong at organized crime detection; produces intuitive "evidence packets" (paths, clusters).
- Weaknesses: graph construction and identity resolution quality are critical; can be computationally heavy in real-time; needs clear interpretability mapping.
- Best for: mule networks, collusive merchants, synthetic identity rings, cross-channel laundering patterns.

Practical note: Many high-performing programs use Pattern B + D together: shared features and a graph engine, with domain models retained as specialized heads.

5.2. Detection Methods

A converged system typically uses a layered detection strategy—no single method covers all typologies with acceptable precision.

5.2.1. Rules

- Encode known typologies, policy constraints, and hard controls (e.g., sanctions rules, impossible travel, risky beneficiary setup patterns).
- Strengths: transparent, auditable, fast.
- Risks: brittle under adversarial adaptation; high false positives if not tuned and segmented.

5.2.2. Supervised ML

- Predict outcomes such as fraud loss, scam likelihood, mule probability, or suspicious activity disposition.
- Strengths: strong precision/recall with good labels; supports nuanced interactions.
- Risks: label leakage, delayed outcomes (AML), bias concerns, monitoring complexity.

5.2.3. Anomaly Detection

- Detect deviations from expected behavior per customer/segment/peer group.
- Strengths: useful for emerging typologies and sparse labels.
- Risks: can be noisy; requires good baselines and suppression logic to avoid alert floods.

5.2.4. Graph Analytics

- Detect organized behavior: rings, hubs, pass-through networks, collusive clusters.
- Strengths: aligns with cross-domain typologies; reveals hidden structure.
- Risks: depends on entity resolution; can be compute-intensive; needs explainable outputs.

5.2.5. Temporal Models and Streaming Considerations

Convergence increases the need to model **time** explicitly:

- Sequence/temporal models: capture event ordering (onboarding → beneficiary add → burst transfers → cash-out).
- Streaming features: maintain rolling windows (5m/1h/1d/30d) for velocity, churn, and network change.
- Concept drift handling: criminals adapt quickly; monitor drift in feature distributions and outcome rates by typology/channel.
- Cold start controls: new customers/accounts lack history—use segment priors, network context, and onboarding signals.

5.3. Alert Generation Strategy

5.3.1. Joint Thresholding and Prioritization

In a converged environment, independent thresholds create duplicates and inconsistent priority. Instead:

- Maintain domain scores (fraud loss risk, AML suspicion, network risk) and a joint priority score that weights harm + suspicion + network impact.
- Calibrate thresholds by action type, not just by domain:
  - *Real-time actions:* step-up auth, hold, block, beneficiary delay.
  - *Investigative actions:* queue to case, request information, enhanced due diligence.

5.3.2. Alert De-Duplication

A common failure mode is "one behavior, many alerts." Converged systems should:

- deduplicate by entity + typology + time window, and
- roll multiple triggers into a single investigative packet.

5.3.3. "Risk Events" vs "Alerts" Separation

A useful design is to separate raw detections from human-facing alerts:

- Risk events: atomic signals emitted by detectors (rule hit, model spike, graph motif detected).
- Alerts: curated bundles of risk events that meet prioritization and routing logic, with evidence attached.

This separation reduces alert fatigue and preserves a full audit trail of what was detected, even if it was suppressed or bundled.

5.4. Explainability and Audit Trails

Converged detection must satisfy both operational explainability (fast action) and compliance defensibility (audit-ready rationale).

- Local explanations: per decision reason codes (top features, thresholds crossed, sequence triggers).
- Typology mapping: every alert should map to one or more typologies with named indicators ("beneficiary churn + pass-through velocity consistent with mule activity").
- Graph evidence packets: include:
  - key paths (entity → entity relationships),
  - cluster membership and size,
  - flow summaries (fan-in/fan-out, pass-through timing),
  - link confidence and timestamps.

- Audit trail requirement: store feature snapshots, model/rule versions, graph state references, and investigator actions so the decision can be reproduced later.

5.5. Model Risk Management

Convergence increases model complexity and therefore governance expectations.

- Validation: out-of-sample testing by typology, channel, and segment; stress testing for adversarial behavior.
- Monitoring: performance drift (AUC/precision/recall), alert volumes, investigation outcomes, false positive rates, and operational SLAs.
- Bias checks: fairness assessments where relevant (especially for identity signals and behavioral proxies); review disparate impact in friction and false positives.
- Drift detection: feature distribution drift, label drift, and graph-structure drift (sudden growth of clusters).
- Champion/challenger: controlled experimentation to introduce new models without destabilizing operations; staged rollout and rollback plans.
- Change control: explicit versioning and documentation tying model changes to observed threat evolution and KPI impact.

Methods Comparison:

Table 3: Comparison of Fraud Detection Approaches Across Performance and Deployment Dimensions

Method	Typical lift	Interpretability	Latency fit	Deployment risk	Best use cases
Rules/scenarios	Medium (known patterns)	High	Excellent	Low–Medium (brittle)	sanctions, hard controls, clear typologies
Supervised ML	High (with labels)	Medium	Good–Excellent	Medium	fraud loss risk, scam likelihood, prioritization
Anomaly detection	Medium (emerging)	Medium	Good	Medium–High (noisy)	novel behaviors, peer group deviations
Graph analytics	High (organized crime)	Medium–High (with packets)	Medium (real-time harder)	Medium	mule rings, collusion, layering networks
Temporal/sequence models	High (ordered behaviors)	Medium	Medium	Medium–High	ATO-to-cashout chains, scam-to-onramp sequences

6. Unified Case Management and Operations

Detection convergence only delivers value if investigations converge too. Unified case management turns cross-domain signals into coherent decisions, reduces duplicated effort, and produces defensible narratives that satisfy both customer-protection goals (fraud) and regulatory obligations (AML). This section outlines why siloed casework fails, defines a unified case model, proposes an end-to-end workflow, and describes deduplication, investigator productivity, and feedback loops.

6.1. Limitations of Siloed Case Management

When fraud and AML teams work in separate case tools (or separate queues within a tool), three systemic problems emerge:

- Duplicate cases for the same entity. The same customer, account, device, or beneficiary can trigger both fraud and AML alerts. Separate queues create parallel investigations, repeated evidence collection, and redundant customer outreach.
- Inconsistent decisions and treatment. One team may close a case as “customer error” or “fraud victim,” while the other may conclude “suspicious activity monitor or report.” These inconsistencies degrade risk posture and create governance exposure.
- Fragmented narratives and weak disruption. Fraud cases often focus on the initiating event (ATO, scam), while AML cases focus on fund flows after the fact. Without a single timeline, investigators can’t connect upstream triggers (identity compromise, social engineering) to downstream laundering (mule dispersal, cash-out). Criminal networks benefit from this fragmentation.

`

Operationally, silos drive alert fatigue, slow time-to-disposition, and missed opportunities to identify and disrupt mule networks and repeat actors.

6.2. Unified Case Model

A unified case model standardizes objects and relationships so both fraud and AML investigations share the same “truth.” At minimum, a converged platform should support these core objects:

- Entity: the canonical subject (person, business, account, device, merchant, counterparty) with identity-resolution context and risk profile.
- Alert: a curated investigative unit generated from one or more risk events; includes typology labels, priority, and routing metadata.
- Event: atomic occurrences (beneficiary added, login anomaly, transfer initiated, refund issued, sanctions hit resolved).
- Network: graph context linking entities (clusters, paths, relationships, flow summaries).
- Evidence: supporting artifacts and signals (feature snapshots, documents, screenshots, call logs, communications, transaction details) with provenance.
- Actions: investigator and system actions (hold funds, request information, file SAR, restrict account, close as false positive) with timestamps and author.
- Disposition: standardized outcomes (confirmed fraud, suspected laundering, monitoring continued, false positive, etc.) with reason codes and confidence.
- Case: the container that links the above objects into a single timeline and narrative.
- Design principle: the "case" is not just a folder—it is a versioned decision record that can be reproduced later (inputs, model/rule versions, graph state reference, and human actions).

6.3. Workflow Design

A converged workflow should preserve domain-specific obligations while eliminating duplication. A practical lifecycle is:

Intake:

- Alerts enter from fraud models/rules, AML monitoring, sanctions screening, graph engine, and anomaly detectors.
- Each alert is tagged with: entity ID(s), typology label(s), urgency, suggested action types, and evidence pointers.

Triage:

- Joint prioritization uses harm + suspicion + network impact.
- Deduplication merges related alerts into a single case (see 6.4).
- Routing assigns to the right queue: real-time interdiction support, fraud investigation, AML investigation, or joint complex-case team.

Investigation:

- Investigator views a unified timeline: identity/onboarding signals, authentication events, transactions, counterparty history, graph connections.
- Playbooks mapped to typologies guide checks and evidence collection.

Escalation:

- Escalate based on thresholds: high-value loss risk, network indicators, potential insider risk, sanctions escalation, or multi-jurisdiction exposure.
- Include specialized review: legal/compliance sign-off, model risk escalation, or financial intelligence unit (FIU) review.

Reporting:

- Generate required outputs: SAR narratives, internal suspicious activity logs, chargeback packages, law enforcement referrals (as applicable).
- Ensure evidence provenance and audit trails are attached.

Closure:

- Close with disposition, reason codes, confidence, and actions taken.
- Document customer impact and remediation steps where relevant.

Feedback to Models:

- Convert dispositions into labeled outcomes and retraining signals (see 6.6).
- Log suppressions, duplicates, and investigator overrides as quality signals for tuning.

6.4. Case Linking and Deduplication

Case linking is a primary ROI driver of unified case management. The goal is to prevent "many alerts, many cases" for the same underlying behavior.

Deduplication strategies:

- Entity-centric dedup: merge alerts that share the same primary entity within a time window (e.g., 24–72 hours), especially if typology labels overlap.
- Network-centric linking: if alerts land on different entities but belong to the same detected cluster (mule ring, collusive merchants), link them into:
  - one master investigation with child cases, or
  - one case with multiple subjects (depending on governance).
- Behavioral sequence linking: connect alerts that form a chain (onboarding anomaly → beneficiary churn → pass-through → cash-out).
- Evidence reuse: shared evidence objects reduce repeated work (same device fingerprint, same counterparty path, same document).

Merging logic should preserve auditability:

- Keep original alerts as immutable records ("risk events" history).
- Store the merge rationale: shared entity keys, link confidence, time overlap, and graph evidence.
- Allow "unmerge" with versioned changes if investigators discover false linkage.

6.5. Investigator Experience and Productivity

A unified case tool should not feel like "more data." It should feel like less work and better context.

Key experience features:

- Role-based queues: fraud queue, AML queue, complex-network queue, sanctions escalation queue with shared visibility into linked cases.
- SLAs and workload balancing: prioritize by customer harm, suspicion, network impact, and regulatory deadlines.
- Typology-mapped playbooks: each typology provides:
  - checks to perform,
  - evidence checklist,
  - recommended actions,
  - narrative template elements (for SAR/fraud reporting),
  - common false positive patterns.
- One-click evidence packets: automatically compile:
  - key transactions and timelines,
  - entity graph snapshot and paths,
  - top reason codes and feature contributions,
  - Identity resolution confidence summary.
- Decision consistency controls: standardized dispositions and reason codes reduce variability across teams.

Productivity metrics typically improve when investigators spend less time gathering evidence and more time making decisions.

6.6. Feedback Loops

Convergence enables richer learning signals—if outcomes are captured consistently.

Label Capture (Aligned Across Fraud + AML)

- Confirmed fraud (type + loss amount + recovery status)
- Scam victim confirmed (where applicable)
- Account takeover confirmed
- Mule activity suspected/confirmed
- SAR filed / suspicious activity reported
- Monitoring continued (insufficient evidence but elevated risk)
- False positive / benign explanation
- Policy action taken (account restrictions, offboarding, enhanced due diligence)

How Feedback Improves Detection

- Supervised learning: dispositions become labels (with confidence and time-to-label).
- Rule tuning: false positives feed rule suppression lists and segmentation improvements.
- Graph model learning: confirmed clusters update "known bad" seed sets and edge-weighting strategies.
- Alert quality monitoring: track which detectors produce high-conversion vs low-conversion alerts.

- Human override learning: investigator overrides provide high-value signals on model blind spots.

Critical governance note: AML "suspicion" labels are not always binary truth; store confidence, rationale, and decision basis to avoid contaminating training.

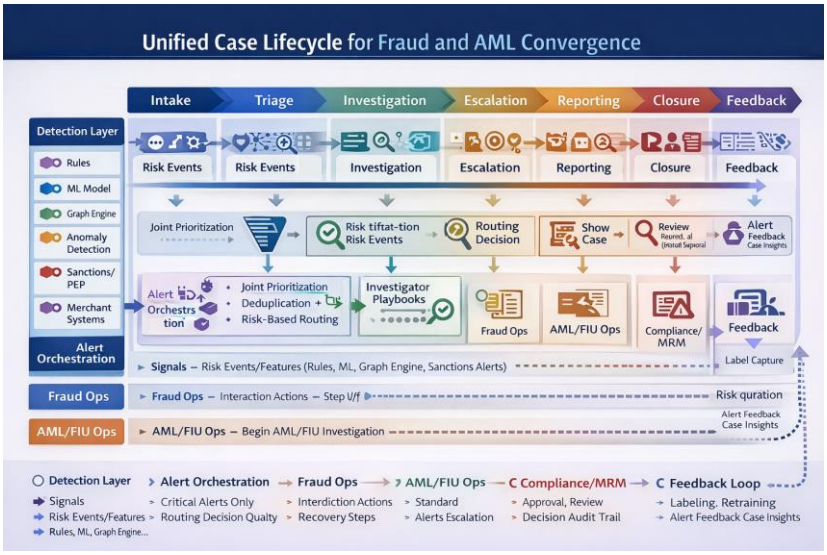

Fig 2: Unified Case Lifecycle Framework for Fraud and AML Operations

7. Evaluation Framework

Evaluating fraud–AML convergence requires a multi-objective lens: improving one outcome (e.g., fraud loss prevented) must not degrade others (e.g., AML defensibility, customer friction, or regulatory posture). This framework defines success, proposes metrics across models and operations, outlines experimental designs and baselines, and provides an error-analysis approach grounded in typologies.

7.1. What "Success" Means (Multi-Objective)

A converged program succeeds when it measurably improves outcomes across three scorecard fraud, AML, and shared operations without creating unacceptable tradeoffs.

Fraud success

- Loss prevented / avoided: reduced gross fraud losses and improved recoveries.
- Chargeback rate reduction: fewer chargebacks and disputes driven by fraud and abuse.
- Customer friction minimized: fewer false declines, fewer unnecessary step-ups, improved approval/conversion.

AML success

- SAR conversion rate improvement: higher proportion of alerts leading to SAR (or equivalent) where appropriate.
- Reduction in regulatory findings: fewer issues related to coverage gaps, backlogs, or insufficient documentation.
- Narrative quality uplift: more complete, consistent narratives with clearer typology mapping and evidence.

Shared operations success

- Alert volume reduction: fewer low-quality alerts and better suppression/bundling.
- Time-to-disposition improvement: faster case closure with consistent outcomes.
- Duplication rate reduction: fewer duplicate investigations for the same entity/network.

Key idea: convergence should shift effort from "more alerts" to "better cases," increasing true positive yield per investigator hour.

7.2. Metrics (Model + Workflow)

A credible evaluation must include both detection metrics (how well models/rules rank risk) and workflow metrics (how efficiently the organization turns signals into decisions).

Model / Scoring Metrics

- Precision / Recall: especially at operationally relevant thresholds (top-k alerts per day).
- PR-AUC (precision–recall AUC): preferred over ROC-AUC in imbalanced problems.
- Calibration: how well score probabilities match observed outcomes (important for thresholding).
- Cost-weighted utility: incorporate asymmetric costs:
    - fraud false negatives = losses and customer harm,
    - fraud false positives = friction and revenue leakage,
    - AML false negatives = regulatory risk and undetected laundering,
    - AML false positives = investigator overload and narrative dilution.

Alert and Case Workflow Metrics

- Alert-to-case ratio: how many alerts become one case after dedup/bundling (lower is better if true-positive yield holds).
- Case cycle time: median and tail (p90/p95) time from intake to disposition.
- Investigator touches: number of human actions per case (lower is better if quality holds).
- Backlog and SLA compliance: time-to-first-touch and time-to-closure versus targets.
- Duplication rate: share of cases reopened/created for entities already under investigation.

Network-Level Metrics (For Graph-First or Network-Aware Systems)

- Community detection stability: do clusters remain consistent as new data arrives (robustness against noise)?
- Ring disruption outcomes: number/size/value of mule rings disrupted (e.g., accounts restricted, flows blocked, beneficiaries removed).
- Network recall proxy: share of known-bad entities connected/covered by detected clusters (requires seed sets and careful governance).
- Interdiction leverage: downstream value prevented by disrupting a hub (impact-weighted measure).

7.3. Experimental Design

A strong evaluation blends offline rigor with controlled online rollout.

Offline Back testing (Historical Replay)

- Replay historical streams to compare:
- separate systems vs converged triage vs unified models.
- Use time-based splits (train on past, test on future) to avoid leakage.
- Evaluate per typology and per channel (instant payments, cards, crypto on/off-ramp, merchant refunds).

Online A/B Testing or Phased Rollout (With Guardrails)

Where feasible:

- A/B tests: randomize at customer/account/segment level (carefully to avoid network contamination).
- Phased rollout: start with low-risk segments, shadow mode scoring, then actioning.
- Guardrails:
    - cap friction actions (step-ups/blocks),
    - cap alert volume,
    - monitor adverse customer impact and operational backlog.

Scenario-Based Evaluation Per Typology

Because convergence is typology-driven:

- build test suites for priority typologies (mule networks, ATO→cashout chains, synthetic onboarding, structuring linked to fraud proceeds).
- measure performance on each scenario with consistent labels and evidence requirements.

7.4. Baselines

Define baselines that isolate where value is coming from:

1. Baseline 0: Current state
    - Separate fraud stack + separate AML monitoring + separate case management.
2. Baseline 1: Shared data only
    - Unified data lake/graph exists, but detection and workflows remain independent.
3. Baseline 2: Shared data + unified triage
    - Joint prioritization, deduplication, and unified case management; domain models still separate.

4. Baseline 3: Fully unified detection
- Shared features + ensemble OR multi-task model OR graph-first risk engine feeding both domains.

This sequence helps answer: Is improvement driven by data integration, workflow unification, or modeling convergence?

7.5. Error Analysis

Error analysis should be structured around archetypes and typologies, not just aggregate metrics.

False Positive Archetypes (Examples)
- Shared household devices / small business shared terminals: multiple legitimate users appear linked.
- Payroll / gig payments / marketplace payouts: fan-out patterns that resemble dispersal.
- Seasonal spikes: holiday shopping, tuition payments, travel patterns causing velocity anomalies.
- Legitimate crypto activity: frequent on/off-ramp behavior by traders or businesses.
- Refund-heavy legitimate merchants: specific verticals (e.g., travel) with high refund ratios.

For each archetype: identify which indicators triggered, what suppressions/segmentations would help, and how to preserve recall for true mule/collusion cases.

False Negative Archetypes (Examples)
- Low-and-slow laundering: small transactions over long horizons, avoiding velocity triggers.
- Cross-channel splits: value moved via multiple rails (cards + instant payments + cash withdrawals) to evade single-channel monitors.
- Newly established mule rings: short-lived accounts with minimal history and quick cash-out.
- Identity resolution failures: weak linkage hides network structure (missed device/account/counterparty connections).
- Adversarial adaptation: timing randomization, rotating devices, beneficiary hopping.

Tie false negatives back to data gaps (missing telemetry), IR weakness (link confidence), and model limitations (poor long-horizon features).

Table 4: KPI Definitions, How Measured, Expected Direction

KPI	Definition	How measured	Expected direction with convergence
Fraud loss prevented	Value of losses avoided via blocks/holds/recovery	(Estimated prevented loss + recovered) vs control	↑ increase
Chargeback rate	Disputes per transaction volume	chargebacks / total txns (by channel)	↓ decrease
Customer friction	Unnecessary step-ups/false declines	false declines %, step-up rate, complaint rate	↓ decrease
SAR conversion rate	SARs per alerts (or per cases)	SAR count / alert count (or / case count)	↑ increase (quality-adjusted)
Narrative quality	Completeness + typology clarity	review rubric score; QA pass rate; rework rate	↑ increase
Regulatory findings	Issues from audits/exams	count/severity over period	↓ decrease
Alert volume	Total alerts generated	alerts/day (normalized by volume)	↓ decrease
Alert-to-case ratio	Alerts bundled into cases	alerts / cases	↓ decrease
Case cycle time	End-to-end time to close	median and p90/p95 hours/days	↓ decrease
Duplication rate	Duplicate cases for same entity/network	% of cases with overlap in entity graph	↓ decrease
Investigator touches	Human actions per case	actions logged per case	↓ decrease (while outcomes hold)
Mule-ring disruption	Disruption outcomes	rings detected; hubs removed; downstream value prevented	↑ increase
Model PR-AUC	Ranking quality under imbalance	PR-AUC per typology/channel	↑ increase
Cost-weighted utility	Net benefit under asymmetric costs	utility = benefits − costs (defined weights)	↑ increase

8. Conclusion

Fraud–AML convergence is increasingly necessary because modern financial crime rarely respects organizational boundaries. The same actors, channels, and enabling capabilities—identity compromise, mule recruitment, scam-driven transfers, merchant abuse, and crypto on/off-ramps create end-to-end pipelines that produce both customer harm and laundering risk. As a result, operating separate fraud and AML detection stacks and separate investigative workflows leads to duplicated effort, inconsistent entity risk views, fragmented narratives, and missed opportunities to disrupt networks early.

This paper has argued that effective convergence starts with shared typologies and a common mapping from narrative behaviors to measurable indicators. A layered typology framework (Actor → Capability → Channel → Technique → Indicator → Outcome) provides a practical "translation layer" between investigators, modelers, and governance stakeholders. Building on that foundation, data integration becomes the enabling infrastructure: unifying customer/KYC, transactions, devices, counterparties, merchants, sanctions/PEP, and (where appropriate) adverse media into an entity-centric graph supported by robust identity resolution, data lineage, and privacy-by-design controls.

On the detection side, convergence does not require a single "one-size-fits-all" model. Institutions can adopt pragmatic architectural patterns from shared data with separate models, to shared feature ensembles, to multi-task learning, to graph-first entity risk engines so long as the output is coherent: consistent scoring semantics, explainable reason codes, and defensible audit trails. Crucially, this work highlighted that converged detection must be paired with unified case management, where related alerts are deduplicated and linked into a single investigation, evidence is captured once and reused, dispositions are standardized across domains, and investigator workflows are mapped directly to typologies.

Finally, the paper proposed an evaluation framework that treats success as multi-objective balancing fraud loss prevention and customer experience with AML reporting quality and regulatory defensibility, while improving operational efficiency through reduced alert volumes, faster case cycle times, and fewer duplicate investigations. This is essential because convergence is not merely a technical integration; it is a transformation of governance, operating rhythms, and accountability.

In practice, the path forward is iterative: begin with shared typologies and a unified data/identity foundation, implement unified triage and case management to eliminate duplication, and then mature toward deeper model convergence and graph-first risk engines. Institutions that pursue this roadmap can move from fragmented monitoring to intelligence-led financial crime operations detecting illicit behavior earlier, responding more consistently, and disrupting networks more effectively, while maintaining the explainability, auditability, and privacy controls required in regulated environments.

References

[1] Basel Committee on Banking Supervision. (2017). *Sound management of risks related to money laundering and financing of terrorism*. Bank for International Settlements.

[2] Board of Governors of the Federal Reserve System, Federal Deposit Insurance Corporation, Financial Crimes Enforcement Network, National Credit Union Administration, & Office of the Comptroller of the Currency. (2021). *Answers to frequently asked questions regarding suspicious activity reporting and other anti-money laundering considerations* (Jan 19, 2021).

[3] Council of the European Union. (2024, May 30). *Anti-money laundering: Council adopts package to strengthen EU rules*.

[4] European Banking Authority. (2021). *Guidelines on money laundering and terrorist financing risk factors* (EBA/GL/2021/02).

[5] Routhu, K. K. (2019). Conversational AI in Human Capital Management: Transforming Self-Service Experiences with Oracle Digital Assistant. *International Journal of Scientific Research & Engineering Trends*, *5*(6).

[6] Financial Action Task Force. (2020). *Guidance on digital identity*.

[7] Financial Action Task Force & Egmont Group of Financial Intelligence Units. (2020). *Trade-based money laundering: Trends and developments*.

[8] Federal Reserve. (2011). *Supervisory guidance on model risk management* (SR 11-7).

[9] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 (General Data Protection Regulation). (2016). *Official Journal of the European Union*.

[10] Routhu, K. K. (2019). Hybrid machine learning architecture for absence forecasting within Oracle Cloud HCM. *KOS Journal of AIML, Data Science, and Robotics*, *1*(1), 1-5.

[11] KPMG. (2024). *Money mules: FinCrime's Trojan horse unveiled*.

[12] Financial Intelligence Centre. (2024). *Financial crime insights: Money mules*.

[13] United Arab Emirates Financial Intelligence Unit. (2024). *Financial crime typologies in the financial sector* (Jan 2024).

[14] Europol. (2023, December 4). *Paper trail ends in jail time for 1 013 money mules*.

[15] Europol. (2024). *Consolidated annual activity report 2023*.

[16] Google Cloud. (2023, September 26). *HSBC and Google Cloud expand partnership to accelerate innovation and support HSBC's net zero ambitions*.

[17] Routhu, K. K. (2019). AI-Enhanced Payroll Optimization: Improving Accuracy and Compliance in Oracle HCM. *KOS Journal of AIML, Data Science, and Robotics*, *1*(1), 1-5.
[18] Google Cloud. (2023). *Anti-money laundering AI (AML AI)*.
[19] Bakhshinejad, H., Schulte, M., & Gloe, T. (2024). Detecting money laundering with graph convolutional networks. *Proceedings of the 16th International Joint Conference on Computational Intelligence (IJCCI 2024)* (pp. 37–47). SCITEPRESS. https://doi.org/10.5220/0013071700003837
[20] Oztas, A. (2024). Transaction monitoring for money laundering prevention: A supervised machine learning approach. *Journal of Money Laundering Control*. https://doi.org/10.1108/JMLC-02-2024-0020
[21] Weber, M., Domeniconi, G., Chen, J., Weidele, D. K. I., Bellei, C., Robinson, T., & Leiserson, C. E. (2019). *Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics* (arXiv:1908.02591). arXiv.
[22] Kranthi Kumar Routhu. (2020). Intelligent Remote Workforce Management: AI, Integration, and Security Strategies Using Oracle HCM Cloud. KOS Journal of AIML, Data Science, and Robotics, 1(1), 1–5. https://doi.org/10.5281/zenodo.17531257
[23] Routhu, K. K. (2020). Strategic Compensation Equity and Rewards Optimization: A Multi-cloud Analytics Blueprint with Oracle Analytics Cloud. *Available at SSRN 5737266*.
[24] Dal Pozzolo, A., Caelen, O., Johnson, R. A., & Bontempi, G. (2014). Calibrating probability with undersampling for unbalanced classification. In *2014 IEEE Symposium Series on Computational Intelligence* (pp. 159–166). IEEE. https://doi.org/10.1109/SSCI.2014.33
[25] Lopez-Rojas, E. A., Elmir, A., & Axelsson, S. (2016). PaySim: A financial mobile money simulator for fraud detection. In *Proceedings of the European Modeling and Simulation Symposium (EMSS 2016)*.
[26] Jullum, M., Løland, A., Huseby, R. B., Ånonsen, G., & Lorentzen, J. (2020). Detecting money laundering transactions with machine learning. *Journal of Money Laundering Control, 23*(1), 173–186. https://doi.org/10.1108/JMLC-01-2019-0007
[27] Routhu, K. K. (2018). Reusable Integration Frameworks in Oracle HCM: Accelerating Enterprise Automation through Standardized Architecture. *International Journal of Scientific Research & Engineering Trends*, *4*(4).
[28] Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.
[29] Grover, A., & Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 855–864). https://doi.org/10.1145/2939672.2939754
[30] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1135–1144). https://doi.org/10.1145/2939672.2939778
[31] Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*.
[32] Ruder, S. (2017). *An overview of multi-task learning in deep neural networks* (arXiv:1706.05098). arXiv.
[33] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys, 46*(4), 1–37. https://doi.org/10.1145/2523813
[34] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357.
[35] Mamidala, J. V., Enokkaren, S. J., Attipalli, A., Bitkuri, V., Kendyala, R., & Kurma, J. (2023). Machine Learning Models Powered by Big Data for Health Insurance Expense Forecasting. *International Research Journal of Economics and Management Studies IRJEMS*, *2*(1).
[36] Bitkuri, V., Kendyala, R., Kurma, J., Enokkaren, S. J., & Mamidala, J. V. (2023). Forecasting Stock Price Movements With Deep Learning Models for time Series Data Analysis. *Journal of Artificial Intelligence & Cloud Computing. SRC/JAICC-531. DOI: doi. org/10.47363/JAICC/2023 (2)*, *489*, 2-9.
[37] Singh, A. A. S. S., Mania, V., Kothamaram, R. R., Rajendran, D., Namburi, V. D. N., & Tamilmani, V. (2023). Exploration of Java-Based Big Data Frameworks: Architecture, Challenges, and Opportunities. Journal of Artificial Intelligence & Cloud Computing, 2(4), 1-8.
[38] Routhu, K. K. (2023). AI-driven succession planning in Oracle HCM Cloud: Building resilient leadership pipelines through predictive analytics. *International Journal of Science, Engineering and Technology*, *11*(5).
[39] Tamilmani, V., Namburi, V. D., Singh Singh, A. A., Maniar, V., Kothamaram, R. R., & Rajendran, D. (2023). Real-Time Identification of Phishing Websites Using Advanced Machine Learning Methods. *Available at SSRN 5837142*.
[40] From Fragmentation to Focus: The Benefits of Centralizing Procurement. (2023). International Journal of Research and Applied Innovations, 6(6), 9820-9833. https://doi.org/10.15662/
[41] Routhu, K. K. (2023). Embedding fairness into the digital enterprise, data driven DEI strategies with Oracle HCM Analytics. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, *9*(8), 266-274.

[42] Routhu, K. K. (2023). AI-driven skills forecasting in Oracle HCM Cloud: From static competencies to predictive workforce design. *International Journal of Science, Engineering and Technology*, *11*(1).
[43] Vattikonda, N., Gupta, A. K., Polu, A. R., Narra, B., Buddula, D. V. K. R., & Patchipulusu, H. H. S. (2022). Blockchain Technology in Supply Chain and Logistics: A Comprehensive Review of Applications, Challenges, and Innovations. *International Journal of Emerging Trends in Computer Science and Information Technology*, *3*(3), 72-80.
[44] Attipalli, A., BITKURI, V., Mamidala, J. V., Kendyala, R., & KURMA, J. (2022). Empowering Cloud Security with Artificial Intelligence: Detecting Threats Using Advanced Machine learning Technologies. *Available at SSRN 5741263*.
[45] Routhu, K. K. (2022). From RFID to Geofencing: IoT-Enabled Smart Time Tracking in Oracle HCM Cloud. *International Journal of Science, Engineering and Technology*, *10*(4).
[46] Polam, R. M., Kamarthapu, B., Kakani, A. B., Nandiraju, S. K. K., Chundru, S. K., & Vangala, S. R. (2022). Data Security in Cloud Computing: Encryption, Zero Trust, and Homomorphic Encryption. *International Journal of Emerging Trends in Computer Science and Information Technology*, *3*(4), 31-41.
[47] Routhu, K. K. (2022). From Case Management to Conversational HR: Redefining Help Desks with Oracle's AI and NLP Framework. *International Journal of Science, Engineering and Technology*, *10*(6).
[48] Polu, A. R., Buddula, D. V. K. R., Narra, B., Gupta, A., Vattikonda, N., & Patchipulusu, H. (2021). Evolution of AI in Software Development and Cybersecurity: Unifying Automation, Innovation, and Protection in the Digital Age. *Available at SSRN 5266517*.
[49] Bitkuri, V., Kendyala, R., Kurma, J., Mamidala, V., Enokkaren, S. J., & Attipalli, A. (2021). Systematic Review of Artificial Intelligence Techniques for Enhancing Financial Reporting and Regulatory Compliance. *International Journal of Emerging Trends in Computer Science and Information Technology*, *2*(4), 73-80.
[50] Attipalli, A., Enokkaren, S., BITKURI, V., Kendyala, R., KURMA, J., & Mamidala, J. V. (2021). Enhancing Cloud Infrastructure Security Through AI-Powered Big Data Anomaly Detection. *Available at SSRN 5741305*.
[51] Singh, A. A. S., Tamilmani, V., Maniar, V., Kothamaram, R. R., Rajendran, D., & Namburi, V. D. (2021). Predictive Modeling for Classification of SMS Spam Using NLP and ML Techniques. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, *2*(4), 60-69.
[52] Kothamaram, R. R., Rajendran, D., Namburi, V. D., Singh, A. A. S., Tamilmani, V., & Maniar, V. (2021). A Survey of Adoption Challenges and Barriers in Implementing Digital Payroll Management Systems in Across Organizations. *International Journal of Emerging Research in Engineering and Technology*, *2*(2), 64-72.
[53] Rajendran, D., Namburi, V. D., Singh, A. A. S., Tamilmani, V., Maniar, V., & Kothamaram, R. R. (2021). Anomaly Identification in IoT-Networks Using Artificial Intelligence-Based Data-Driven Techniques in Cloud Environmen. *International Journal of Emerging Trends in Computer Science and Information Technology*, *2*(2), 83-91.
[54] Attipalli, A., BITKURI, V., KURMA, J., Enokkaren, S., Kendyala, R., & Mamidala, J. V. (2021). A Survey of Artificial Intelligence Methods in Liquidity Risk Management: Challenges and Future Directions. *Available at SSRN 5741342*.
[55] Routhu, K. K. (2021). AI-augmented benefits administration: A standards-driven automation framework with Oracle HCM Cloud. *International Journal of Scientific Research and Engineering Trends*, *7*(3).
[56] Routhu, K. K. (2021). Harnessing AI Dashboards in Oracle Cloud HCM: Advancing Predictive Workforce Intelligence and Managerial Agility. *International Journal of Scientific Research & Engineering Trends*, *7*(6).